

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/152.00.1/Б/ОК25- 2022
	Екземпляр № 1	Арк 69 / 1

## **ЗАТВЕРДЖЕНО**

Науково-методичною радою  
Державного університету  
«Житомирська політехніка»  
протокол від 16 грудня 2022 р.  
№ 13

### **МЕТОДИЧНІ РЕКОМЕНДАЦІЇ для проведення лабораторних робіт з навчальної дисципліни «МІКРОПРОЦЕСОРИ ТА МІКРОКОНТРОЛЕРИ У ІНФОРМАЦІЙНО- ВИМІРЮВАЛЬНІЙ ТЕХНІЦІ»**

для здобувачів вищої освіти освітнього ступеня «бакалавр»  
спеціальності 152 «Метрологія та інформаційно-вимірвальна техніка»  
освітньо-професійна програма «Комп'ютеризовані інформаційно-  
вимірвальні системи»  
факультет комп'ютерно-інтегрованих технологій, мехатроніки і  
робототехніки

кафедра метрології та інформаційно-вимірвальної техніки

Рекомендовано на засіданні  
кафедри метрології та  
інформаційно-вимірвальної  
техніки  
30 серпня 2022р., протокол № 8

Укладач: к.т.н., доцент кафедри метрології та інформаційно-вимірвальної  
техніки ЧЕПЮК Ларіна

Житомир  
2022

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 2

Методичні рекомендації до лабораторних робіт з дисципліни «Мікропроцесори та мікроконтролери у інформаційно-вимірювальній техніці» для студентів спеціальності 152 «Метрологія та інформаційно-вимірювальна техніка»/ Укладач Л.О. Чепюк, – Житомир: ДУ «Житомирська політехніка», 2022. – 69 с.

Укладач: Л.О. Чепюк

Рецензенти:

д.т.н., професор кафедри робототехніки, електроенергетики та автоматизації ім. проф. Б.Б. Самотокіна Кирилович В.А.,  
к.т.н., доцент кафедри робототехніки, електроенергетики та автоматизації ім. проф. Б.Б. Самотокіна Шавурський Ю.О.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 3

## ЗМІСТ

<b>Лабораторна робота № 1</b>	
Керування портами вводу/виводу мікроконтролера	4
<b>Лабораторна робота № 2</b>	
Інтерфейс вводу даних від користувача	12
<b>Лабораторна робота № 3</b>	
Організація взаємодії мікроконтролера з персональним комп'ютером	16
<b>Лабораторна робота № 4</b>	
Керування драйверами та приводами	23
<b>Лабораторна робота № 5</b>	
Створення першого проекту на мові С	32
<b>Лабораторна робота № 6</b>	
Дискретизація та аналогово-цифрове перетворення сигналів	41
<b>Лабораторна робота № 7</b>	
Створення уніфікованих проектів на С для ARM процесорів	64
<b>Лабораторна робота № 8</b>	
Застосування Matlab Simulink для програмування мікроконтролерів STM32F4	57

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 4

## Лабораторна робота № 1

### КЕРУВАННЯ ПОРТАМИ ВВОДУ/ВИВОДУ МІКРОКОНРОЛЕРА

#### 1.1 Мета та основні завдання роботи

Ознайомитись з середовищем розроблення програмного забезпечення Keil MDK для створення, побудови та тестування проектів вбудовуваних систем на мікроконтролерах ARM архітектури фірми NUVOTON, отримати практичні навички програмування мікроконтролерів на асемблері.

#### 1.2 Завдання і порядок виконання роботи

В лабораторній роботі слід створити, побудувати та виконати тестування простого проекту. Завдання виконується в послідовності:

- Запустити майстер для створення проекту та вибрати базу даних файлів NuMicro Cortex M0 Database; вибрати тип мікроконтролера – NUC140VE3CN, додати в проект шаблон програми на асемблері startup\_NUC1xx.s. Докладно покрокові інструкції створення проекту розглянуто у додатку А.
- Побудувати проект та тестувати програму за допомогою симулятора  $\mu$ Vision, зокрема використовуючи засоби Watch, Memory, Dissassembly Listing, Step, Step Out, Step Over, Run, Run to Cursor.
- Ознайомитися з особливостями макету “Nu LB NUC140”, зокрема з’ясувати як з’єднані порти вводу/виводу МК з світлодіодами (рис. В.3) та семисегментними індикаторами (рис. В.4).
- Ознайомитися з теоретичними відомостями (п. 1.3) методичних вказівок.
- Створити файл **reg\_def.s** та внести в нього назви регістрів та їх адреси через директиву EQU
- Написати програму запропонованої структури та виконати кейси.
- Перевірити кожне завдання кейсів на макеті з використанням вбудованого відладчика debugger.
- Уточнити індивідуальне завдання.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 5

## 1.3 Основні теоретичні відомості

### 1.3.1 Організація проекту

За детальним описом створення, побудови, ладнання та симуляції проекту в середовищі Keil MDK зверніться до додатку А. Для того, щоб результати вже виконаних завдань лабораторної роботи не втрачались під час виконання наступних рекомендується кожне завдання проекту організувати у вигляді підпрограм. При цьому виклик вже виконаних завдань можна коментувати.

#### Loop

```

BL case0 ;
;BL case1 ; закоментовано
;BL case2 ; закоментовано

```

#### В Loop

### 1.3.2 Особливості написання програм для мікроконтролерів з архітектурою ARM v6M

Для створення проекту сумісного з макетом, що використовується на лабораторних роботах, необхідно обрати мікроконтролер PIC18F4550. Оскільки в архітектурі процесора ARM-v6M на основі якого створено Cortex-M0 наявні виключення (переривання від апаратних пристроїв), котрі неможливо вимкнути або заборонити (NMI, fault\_exception, SVC), то необхідно навіть у найпростішій програмі на асемблері прописувати таблицю виключень та переривань, яка задає відповідний розподіл пам'яті. Тому доцільно використовувати розроблений виробником мікроконтролерів шаблоном **startup\_NUC1xx.s**. У файлі шаблону виконується базова конфігурація мікроконтролера: задається початкова адреса стеку, розміщується таблиця векторів виключень та переривань мікроконтролера, об'являються назви підпрограм обробки виключень та переривань. Серед інструкцій також використовуються директиви асемблера, такі як: **ALIGN, REA, END, EXPORT, IMPORT, ENTRY, PROC ENDP, IF ENDF, INCLUDE**. За допомогою цих директив виконується розподіл пам'яті, можливо організувати ієрархію проекту, досягти кращої навігації по проекту і краще його розуміння. Далі наведено частину програми **startup\_NUC1xx.s**.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 6

; Vector Table Mapped to Address 0 at Reset

	AREA	RESET, DATA, READONLY	
	EXPORT	__Vectors	
<b>__Vectors</b>	DCD	__initial_sp	; Top of Stack
	DCD	Reset_Handler	; Reset
<b>Handler</b>	DCD	NMI_Handler	; NMI Handler
	DCD	HardFault_Handler	; Hard Fault
<b>Handler</b>	DCD	0	; Reserved
	DCD	0	; Reserved
	DCD	0	; Reserved
	DCD	0	; Reserved
	DCD	0	; Reserved
	DCD	0	; Reserved
	DCD	0	; Reserved
	DCD	0	; Reserved
	DCD	SVC_Handler	; SVCcall
<b>Handler</b>	DCD	0	; Reserved
	DCD	0	; Reserved
	DCD	PendSV_Handler	; PendSV
<b>Handler</b>	DCD	SysTick_Handler	; SysTick
<b>Handler</b>			

; External Interrupts

; maximum of 32 External Interrupts are possible

DCDBOD\_IRQHandler  
DCDWDI\_IRQHandler  
DCDEINT0\_IRQHandler  
DCDEINT1\_IRQHandler  
DCDGPAB\_IRQHandler  
DCDGPCDE\_IRQHandler  
DCDPWMA\_IRQHandler  
DCDPWMB\_IRQHandler  
DCDTMR0\_IRQHandler  
DCDTMR1\_IRQHandler  
DCDTMR2\_IRQHandler  
DCDTMR3\_IRQHandler  
DCDUART02\_IRQHandler  
DCDUART1\_IRQHandler  
DCDSPI0\_IRQHandler  
DCDSPI1\_IRQHandler

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 7

```

DCDSPI2_IRQHandler
DCDSPI3_IRQHandler
DCDI2C0_IRQHandler
DCDI2C1_IRQHandler
DCDCAN0_IRQHandler
DCDDefault_Handler
DCDDefault_Handler
DCDUSBD_IRQHandler
DCDPS2_IRQHandler
DCDACMP_IRQHandler
DCDPDMA_IRQHandler
DCDI2S_IRQHandler
DCDPWRWU_IRQHandler
DCDADC_IRQHandler
DCDDefault_Handler
DCDRTC_IRQHandler

```

### 1.3.3 Налаштування тактового генератора МК

Ініціалізацію тактового генератора слід виконувати відповідно до технічної документації до мікроконтролера NUC140 Technical Reference Manual, TRM\_NUC130\_NUC140(CN)\_Series\_EN\_Rev2.05.pdf (підр. 5.3) [3]. У нашому випадку налаштуємо тактовий генератор на роботу від зовнішнього кварцу з частотою 12 МГц, з включеним помножувачем частоти (PLL) і робочою частотою 48 МГц.

```
;Init Clock Gen for 48 MHz
```

```

LDR      R0, =CLK_BA
LDR      R1, =0x1D; Enable external 4~24 MHz high speed
STR      R1, [R0] ; XTL12M_EN = 1

LDR      R1, =0x0000C22E; IN_DV=0x1, OUT_DV=0x3, FB_DV=0x2E
STR      R1, [R0, #PLLCON] ; PLLFOUT = 48 MHz from PLL

MOVS    R1, #0x3 ; SysTick clock source select HCLK/2
LSLS    R1, R1, #3
MOVS    R2, #0x2
ORRS    R1, R2 ; CLK SRC for CPUCLK, HCLK, PCLK - PLLFOUT
STR      R1, [R0, #CLKSEL0]

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 8

```
MOVS R1, #0 ; HCLK_N = 0, divide by 1
STR R1, [R0, #CLKDIV]; PLLFOUT = 48 MHz from PLL
```

### 1.3.4 Ініціалізація портів вводу/виводу МК

Ініціалізацію портів вводу/виводу (ПВВ) слід виконувати відповідно до технічної документації, підр. 5.5 [3]. Загалом налаштування зводиться до запису конфігуруючих бітів у регістри GPIOx\_PMD та GPIOx\_DOUT, де x – буква, що характеризує назву ПВВ. Регістр GPIOx\_PMD визначає режим роботи ПВВ: вхід, вихід, чи двонаправлений порт. Регістр GPIOx\_DOUT визначає логічне значення на виході ПВВ: лог. 1 або лог. 0.

Зверніть увагу, що виводи мікросхеми МК можуть розділятися з тими чи іншими периферійними пристроями. Тому, необхідно встановити чи не потрібно виконувати додаткові налаштування задля того, щоб використовувати їх за призначенням.

```
;Init GPIO for L
LDR R0, =GP_BA
LDR R1, =0x55000000
STR R1, [R0, R2]

LDR R1, =0x0000F000
LDR R2, =GPIOC_DOUT
STR R1, [R0, R2]
```

### 1.3.5 Ініціалізація системного SysTick таймера

Оскільки тактова частота МК доволі значна (48 МГц) виникає задача формування часових затримок під час реалізації тієї чи іншої функціональності. З цією метою розробники ARM архітектури радять використовувати апаратно вбудовані в МК таймери. Тому з цією метою налаштуємо роботу системного таймеру для формування часового проміжку, що буде рівним 1 мс. Опис налаштувань таймера SysTick приведений у п. 5.2.6 технічної документації [3].

```
;SysTick Timer Init to form 1 ms
LDR R0, =SCS_BA
LDR R1, =47999
STR R1, [R0, #SYST_RVR]
```



Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 9

```

MOVS    R1, #0
STR     R1, [R0, #SYST_CVR]
MOVS    R1, #5
STR     R1, [R0, #SYST_CSR]

```

### 1.3.6 Розблокування регістрів системних налаштувань

За замовчуванням регістри з важливими налаштуваннями заблоковані від випадкового запису в них. У пункті 5.2.5 технічної документації [3] описується регістр та процедура розблокування шляхом запису в REGWRPROT (Register Write-Protection Control Register) послідовності даних “59h”, “16h” “88h”. Приклад реалізації розблокування і потім наступного блокування регістрів є у файлі шаблону **startup\_NUC1xx.s**.

## 1.4 Навчальні приклади

### 1.4.1 Case0 Керування світлодіодами

Схема з'єднання портів вводу/виводу МК з світлодіодами приведена у Nu\_LB-002 Rev 2.0 User's Manual . Для можливості відображення необхідно в циклі ввести довгу затримку(~ 1 сек).

```

LDR     R0, =SCS_BA
Blinky_inner_loop1
Blinky_inner_loop2
        LDR     R2, =0x10000
        LDR     R3, [R0, #SYST_CSR]
        TST     R3, R2
        BEQ     Blinky_inner_loop2
        SUBS   R1, R1, #1
        BNE     Blinky_inner_loop1

```

### 1.4.2 Case1 Формування затримки 500 мс

Для реалізації динамічної індикації використовуються змінні SEG1, SEG2, SEG3, SEG4 для зберігання і наступного виводу даних на індикатори.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 10

```

LDR      R1,    =500
Blinky_inner_loop1
Blinky_inner_loop2
LDR      R2,    =0x10000
LDR      R3,    [R0, #SYST_CSR]
TST      R3, R2
BEQ      Blinky_inner_loop2
SUBS     R1, R1, #1

BNE      Blinky_inner_loop1

```

### 1.4.3 Case0 Керування світлодіодами

Організувати лічильник, значення якого має виводитись на світлодіоди, навчального макету, номери ПБВ слід з'ясувати за схемою електричною принциповою.

### 1.4.4 Case2 Вивід значення лічильника на 7-сегментний індикатор

Необхідно організувати вивід значення лічильника на семисегментний індикатор. Реалізувати динамічну індикацію.

## 1.5 Варіанти індивідуальних завдань

Таблиця 1.1 – Приклади варіантів завдань

№	Завдання	№	Завдання
1	Дин. інд лічильник до 100	7	Дин. інд-я лічильник до 100
2	Процес загрузки світлодіоди	8	Дин. інд-я символів ряду Фібоначі вниз
3	Дин. інд-я біжуча стрічка вліво 4 симв	9	Дин. інд-я лічильник до 100
4	Дин. інд-я декр лічильника 99	10	Дин. інд-я лічильника до 1000
5	Дин. інд-я реверс лічильник 0-100	11	Дин. інд-я реверс лічильник 50-150
6	Дин. інд-я символів ряду Фібоначі	12	Дин. інд біжуча стрічка вправо 4 симв

## Контрольні запитання

1. Навіщо у файлі шаблоні startup\_NUC1xx.s приводиться таблиця векторів.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 11

2. Поясніть що виконує команда **BX LR**.
3. Поясніть що виконує команда **LDR R1, =GP\_BA**, яка адреса у **GP\_BA**.
4. Поясніть яку дію виконує команда **TST R3, R2**.
5. Поясніть процедуру розблокування регістрів системних налаштувань.
6. Які налаштування визначаються регістром **GPIOC\_PMD**?
7. До яких портів вводу/виводу приєднані семи сегментні індикатори?
8. Поясніть яку дію виконує команда **MVNS R3, R3**.
9. Поясніть яку дію виконує команда **STR R3, [R1, R2]**.
10. Виконайте розрахунок та визначіть налаштування тактового генератора на 24 МГц.
11. Поясніть навіщо використовуються директиви асемблера **IMPORT** та **EXPORT**.
12. Поясніть призначення директив асемблера **INCLUDE** та **AREA**.

### Література до лабораторної роботи № 1

1. Yiu J. The Definitive Guide to the ARM Cortex-M0 / J. Yiu . – Newnes, 2011. – 518p. – ISBN: 978-0-12-385477-3. [pp. 43-71]
2. Knaggs P. ARM: Assembly Language Programming / P. Knaggs, S. Welsh. – School of Design, Engineering & Computing, August 31 2004. – 156 p. [pp. 11-22]
3. NUC140 Datasheet EN V3.02 [Електронний документ]. – Режим доступу : <http://www.nuvoton.com/resource-files/DA00-NUC140ENF1.pdf> – Назва з екрану. [pp. 47-151, 177-199].
4. Вбудовані системи <http://radar.kpi.ua/downloads/>

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 12

## Лабораторна робота № 2

### ІНТЕРФЕЙС ВВОДУ ДАНИХ ВІД КОРИСТУВАЧА

#### 2.1 Мета та основні завдання роботи

Ознайомитись з основними особливостями налаштування мікроконтролерів для можливості забезпечення цифрового керування пристроями на прикладі сканування кнопок та матричної клавіатури. Отримати навички ладнання програмного коду з обробкою переривань.

#### 2.2. Завдання і порядок виконання роботи

В лабораторній роботі слід створити, побудувати та виконати тестування проекту, що дасть змогу реалізувати МК інтерфейс взаємодії людини з вбудовуванню системою. Завдання доцільно виконувати у такій послідовності:

- Створити новий проект на основі попередньої лабораторної роботи. У проекті мають бути представлені файли **SystemInit.s** , **main.s** та файл з адресами регістрів **reg\_def.s**. У файлі SystemInit.s має бути наявна ініціалізація тактового генератора, системного таймера SysTick, та портів вводу/виводу, що приєднані до світло діодів. У файлі main.s має бути наявна програма формування затримки в 500 мс.
- Виконати зміни у налаштуваннях тактового генератора та системного таймеру відповідно до п. 2.3.1.
- Виконати додаткову ініціалізацію портів вводу виводу для роботи з RGB світло діодів, зумером та мембранною кнопкою SW\_INT1, відповідно до п.2.3.2.
- Налаштувати переривання INT1 та його обробку від кнопки SW\_INT1, відповідно до п. 2.3.3.

#### 2.3 Основні теоретичні відомості

##### 2.3.1 Зміни у налаштуваннях проекту

Для закріплення практичних навичок у роботі з технічною документацією на мікроконтролер, виконати зміни у налаштуваннях тактового генератора (підр. 5.3 Clock Controller) так, щоб робоча тактову частоту була рівною 24 МГц. При цьому автоматично мають змінитись часові затримки які формуються за допомогою системного таймеру з 1 мс на 2 мс. Зміни мають відобразитись у частоті миготіння світло діодів.

Повернути частоту миготіння світло діодів до попереднього значення за рахунок пере конфігурування системного таймеру (п. 5.2.6 System Timer SysTick [3]) або у підпрограмі затримки в main.s.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 13

### 2.3.2. Ініціалізація портів вводу / виводу (ПВВ)

У даній лабораторній роботі ми будемо досліджувати роботу з ПВВ, що сконфігуровані як входи. Для цього використаємо мембрану кнопку, що приєднана до ПВВ GPIOB15 за схемою, що приведена на рис. 2.1. Відповідно до схеми в неактивному стані на вході МК буде високий рівень (логічна 1), який забезпечує «підтягуючий» (pull-up) резистор 10 кОм. У разі натиснення на кнопку на вході ПВВ формується потенціал землі (логічний 0). Цю особливість роботи схеми необхідно враховувати під час написання програми.

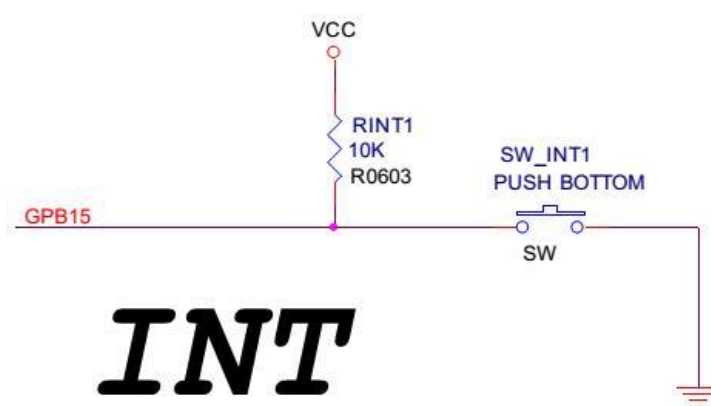


Рис. 2.1 – Вікно керування стимулами

Для наочності роботи з кнопкою доцільно використати додаткові індикатори. На рис. 2.2 приведена схема приєднання до МК RGB світло діода, колір якого буде залежати від сигналів, що подаються GPIOA12, 13 та 14. На схемі рис. 2.3 показано підключення зумера до порту GPIOB11.

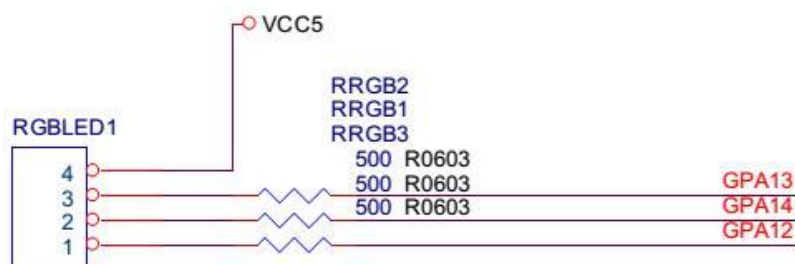


Рис. 2.2 – Вікно керування стимулами

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 14

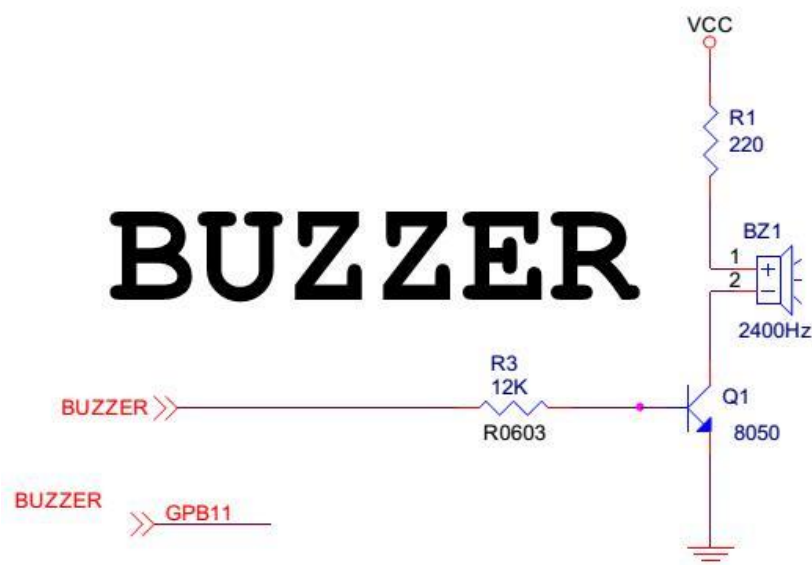


Рис. 2.3 – Вікно керування стимулами

Відповідно до приведених схем до файлу **SystemInit.s** слід додати ініціалізацію GPIOB15 як входу, GPIOB11 та GPIOA12, 13 та 14 як виходів.

### 2.3.3 Налаштування та обробка зовнішнього переривання

Для налаштування переривання з кнопки GPIOB15 слід додатково сконфігурувати регістри GPIOB\_IEN, GPIOB\_IMD. Існує чотири типи переривань, котрі можна налаштувати: переривання у разі появи низького або високого рівня сигналу, переривання за заднім або переднім фронтом сигналу. Якщо обрати джерелом переривання зміну сигналу за фронтом, то додатково можна налаштувати апаратно реалізовану МК схему захисту від «дребезгу контакту» або дії завад під назвою «Debounce» з метою попередження можливої багаторазової реакції на одну подію. Для цього додатково необхідно налаштувати регістр GPIOB\_DBEN, у якому можна обрати джерело тактового сигналу і кількість вибірок за якими приймається рішення про появу відповідного фронту.

Обробник переривань прописаний у таблиці векторів як

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 15

## DCD      EINT1\_IRQHandler

### Контрольні запитання до лабораторної роботи № 2

1. Навіщо у файлі шаблоні startup\_NUC1xx.s приводиться таблиця векторів.
2. Поясніть що виконує команда BX LR.
3. Поясніть що виконує команда LDR R1, =GP\_BA, яка адреса у GP\_BA
4. Поясніть яку дію виконує команда TST R3, R2.
5. Поясніть процедуру розблокування регістрів системних налаштувань.
6. Які налаштування визначаються регістром GPIOC\_PMD?
7. До яких портів вводу/виводу приєднані семи сегментні індикатори?
8. Поясніть яку дію виконує команда MVNS R3, R3.
9. Поясніть яку дію виконує команда STR R3, [R1, R2].
10. Виконайте розрахунок та визначте налаштування тактового генератора на МГц.
11. Поясніть, навіщо використовуються директиви асемблера IMPORT та EXPORT.
12. Поясніть призначення директив асемблера INCLUDE та AREA.

### Література до лабораторної роботи № 2

1. Yiu J. The Definitive Guide to the ARM Cortex-M0 / J. Yiu . – Newnes, 2011. – 518p. – ISBN: 978-0-12-385477-3. [pp. 43-71]
2. Knaggs P. ARM: Assembly Language Programming / P. Knaggs, S. Welsh. – School of Design, Engineering & Computing, August 31 2004. – 156 p. [pp. 11-22]
3. NUC140 Datasheet EN V3.02 [Електронний документ]. – Режим доступу : <http://www.nuvoton.com/resource-files/DA00-NUC140ENF1.pdf> – Назва з екрану. [pp. 47-151, 177-199].
4. Вбудовані системи <http://radar.kpi.ua/downloads/>

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 16

### Лабораторна робота № 3

## ОРГАНІЗАЦІЯ ВЗАЄМОДІЇ МІКРОКОНТРОЛЕРА З ПЕРСОНАЛЬНИМ КОМП'ЮТЕРОМ

### 3.1 Мета та основні завдання роботи

Отримати навички організації взаємодії вбудованих систем з персональним комп'ютером для можливості подальшого оброблення даних в комп'ютері, відображення результатів, а також з метою керування мікроконтролером.

### 3.2 Завдання і порядок виконання роботи

У лабораторній роботі слід створити, побудувати та виконати тестування проекту, що дасть змогу вбудованій системі взаємодіяти з ПК. Завдання доцільно виконувати у такій послідовності:

- Створити новий проект на основі попередньої лабораторної роботи. У проекті мають бути представлені файли **SystemInit.s**, **main.s** та файл з адресами регістрів **reg\_def.s**. У файлі SystemInit.s має бути наявна ініціалізація тактового генератора, системного таймера SysTick, та портів вводу/виводу, що приєднані до світлодіодів. У файлі main.s має бути наявна програма формування затримки в 500 мс
- Виконати ініціалізацію модуля UART, відповідно до п. 3.3.1 та технічної документації.
- Налаштувати програму HyperTerminal та МК для роботи на швидкості 19200 бод.
- Виконати перевірку роботи модуля UART через термінал ПК.
- Уточнити у викладача індивідуальне завдання – *швидкість передачі даних*.
- Виконати навчальні приклади (кейси) передачі та прийому даних з перевіркою на макеті (підр. 3.4).
- Налаштувати переривання INT1 та його обробку від кнопки SW\_INT1, відповідно до п. 2.3.3.

### 3.3 Основні теоретичні відомості

#### 3.3.1 Ініціалізація модуля UART

Ознайомтесь з розділом 5.12 UART Interface Controller документації на мікроконтролер NUC140(CN), зокрема з регістрами, які напряду чи опосередковано впливають на режими роботи модуля. Схема узгоджувача рівнів сигналів між МК та ПК на рис. 3.1.



Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 17

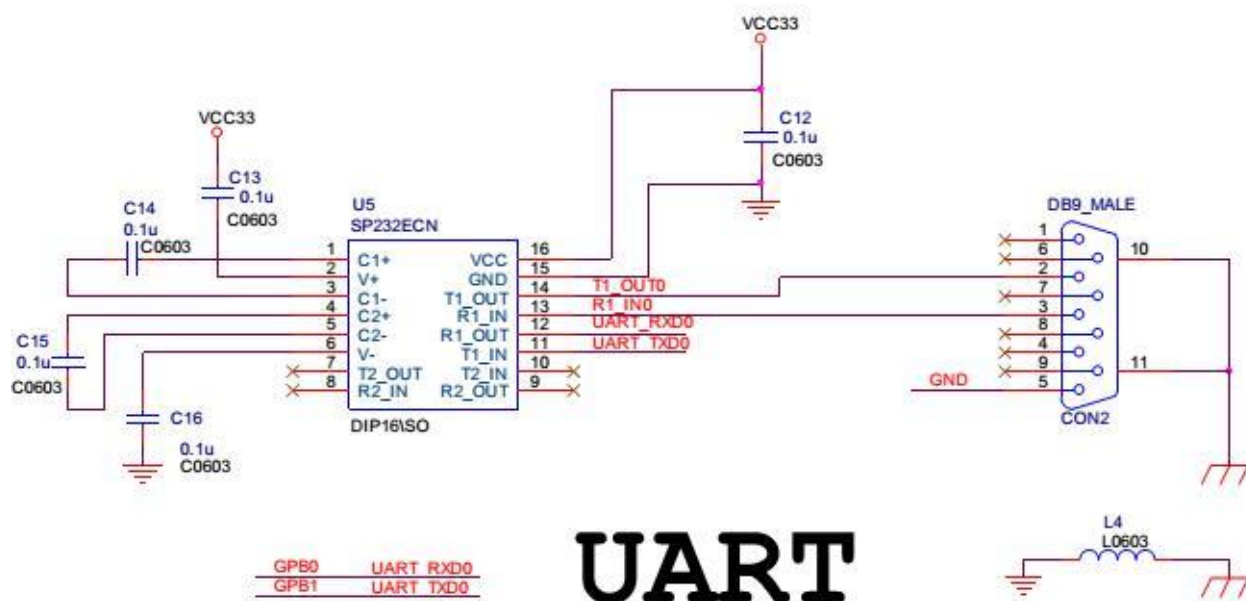


Рис. 3.1 – Схеми інтерфейсу UART до ПК

З схеми на рис. 3.1 видно, що для прийому та передачі даних використовуються ПІВВ GPB0 (UART\_RXD0) та GPB1 (UART\_TXD0), що використовується UART0.

Під час ініціалізації слід посеред різних наявних режимів слід обрати звичайний UART, налаштувати швидкість передачі даних, розрядність даних, кількість стопових біт та функцію перевірки парності. У табл. 3.1. приведена таблиця з регістрами які необхідні для налаштування базової функціональності модуля UART. Також слід налаштувати тактовий сигнал, що буде використовуватися модулем у регістрах CLKSEL1[25:24], CLKDIV[11:8] та APBCLK[16].

Name	Mnemonic	Address	Default Value
GPB_MFP	Multiple Function Pin Control Reg.	GCR_BA+0x34	0x0000_0000
UA_THR	Transmit Holding Register	UART0_BA+0x00	Undefined
UA_RBR	Receive Buffer Register	UART0_BA+0x00	Undefined
UA_LCR	Line Control Register	UART0_BA+0x0C	0x0000_0000
UA_BAUD	Baud Rate Divisor Register	UART0_BA+0x24	0x0F00_0000
UA_FUN_SEL	Function Select Register	UART0_BA+0x30	0x0000_0000

```
;Init UART0
```

```
;Конфігурація виводів МК для роботи з UART
```

```
LDR R0, =GCR_BA ; PB.0 & PB.1 Pin Function Selection MOVS
R2, #0x3 ; 1 - UART0 TXD & RXD STR R2, [R0, #GPB_MFP]
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 18

;Визначення режиму роботи модуля UART0

```

LDR      R0, =UART0_BA
MOVS     R2, #0
STR      R2, [R0, #UA_FUN_SEL]
;Сигнал Reset для UART0
LDR      R0, =GCR_BA      ; SYS->IPRSTC2.UART0_RST = 1;
MOVS     R2, #0x1        ; SYS->IPRSTC2.UART0_RST = 0;
LSLS     R2, #16
STR      R2, [R0, #IPRSTC2]
MOVS     R2, #0x0
STR      R2, [R0, #IPRSTC2]

;Ввімкнення UART0
LDR      R0, =CLK_BA      ; SYSCLK->APBCLK.UART0_EN = 1;
MOVS     R2, #0x1;
LSLS     R2, #16
STR      R2, [R0, #APBCLK]
;Сигнал скидання для буферів Tx та Rx FIFO
LDR      R0, =UART0_BA    ; Set Rx Trigger Level (FCR.RFITL)
MOVS     R2, #0x6;
STR      R2, [R0, #UA_FCR]
;Формат даних: к-ть розр. даних, стопових біт,
LDR      R0, =UART0_BA    ; перевірка на парність
MOVS     R2, #0x3        ; 1 StopBits; No Parity;
STR      R2, [R0, #UA_LCR]
;Set Time-Out (TOR.TOIC)
LDR      R0, =UART0_BA
MOVS     R2, #0x1;
LSLS     R2, #12        ; TX Delay Time Value
STR      R2, [R0, #UA_TOR]
;Налаштування тактового сигналу для модуля UART0
LDR      R0, =CLK_BA      ; SYSCLK->CLKSEL1.UART_S = 0;
LDR      R1, =0x00000000  ; XTL12M to UART
STR      R1, [R0, #CLKSEL1]
;Mode 2, Baud rate = UART_CLK / (A+2), A must >=3

LDR      R0, =UART0_BA    ; Baud rate = 115200
LDR      R2, =0x3f000066  ; DIV_X_EN=1 ; DIV_X_ONE=1 ; A=0x66

STR      R2, [R0, #UA_BAUD]

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 19

### 3.3.2 Налаштування терміналу на ПК

Налаштування терміналу комп'ютера виконується наступним чином: необхідно запустити програму HyperTerminal (Windows XP: Пуск > Программи>Стандартные>Связь>HyperTerminal; Windows 7: встановлюється окремо). При першому запуску потрібно виконати налаштування, зокрема обрати порт COM1, COM2 до якого під'єднаний МК (рис. 3.2). Встановити швидкість передачі даних та їх формат, а саме кількість стопових біт, розрядність даних (рис. 3.3). Дані налаштування повинні співпадати з налаштуваннями послідовного порта USART МК. Вікно терміналу (HyperTerminal) показано на рис. 3.4.

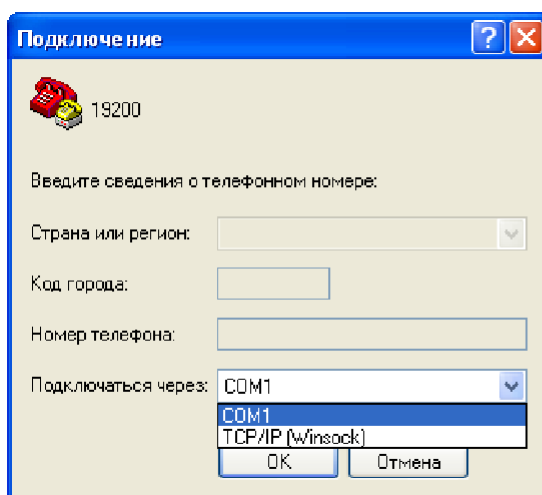


Рис. 3.2 – Вибір послідовного порта

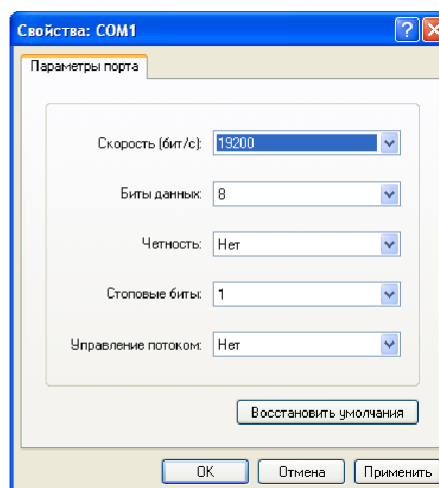


Рис. 3.3 – Швидкість

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 20

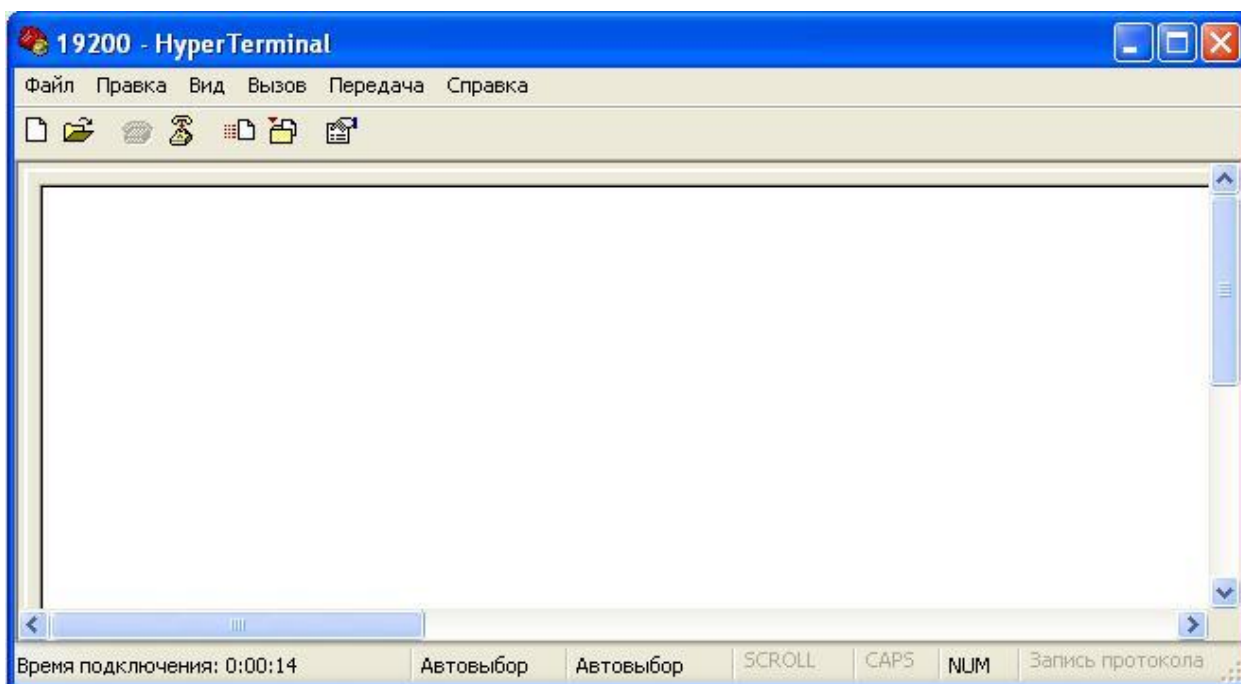


Рис. 3.4 – Вікно HyperTerminal після проведення налаштувань

### 3.3.3 Налаштування та обробка зовнішнього переривання

Для налаштування переривання від модуля UART слід сконфігурувати регістри NVIC та дозволити переривання за вектором UART02. Подія за якою буде формуватись переривання визначається в регістрі UA\_IER. У разі формування переривання під час прийому даних доцільно формувати переривання за подією Receive Data Available (RDA\_IEN).

Обробник переривань слід прописувати за посиланням з таблиці векторів, зокрема

`UART02_IRQHandler`

## 3.4 Навчальні приклади

### 3.4.1 Case0 Перевірка передачі даних послідовним портом

Програма першого кейсу дозволяє перевірити чи вірно налаштований послідовний порт для передачі даних, а також термінал комп'ютера для прийому даних.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 21

```

LDR      R0, =UART0_BA
LDR      R2, =0x10000000
Wait_TE_FLAG
LDR      R3, [R0, #UA_FSR]
TST      R3, R2
BEQ      Wait_TE_FLAG
LDR      R4, ='0'
STR      R4, [R0, #UA_THR]
LDR      R4, ='k'
STR      R4, [R0, #UA_THR]
LDR      R4, ='!'
STR      R4, [R0, #UA_THR]
LDR      R4,='\n'
STR      R4, [R0, #UA_THR]
LDR      R4,='\r'
STR      R4, [R0, #UA_THR]

```

### 3.4.2 Case1 Передавання даних у ASCII коді

Програма даного кейсу дозволяє передавати у ASCII коді значення лічильника до 10

```

case1
    ADDS    R5, #1      ; лічильник до 10
    MOVS    R4, #0xA
    CMP     R5, R4
    BNE    skip
    MOVS    R5, 0
Skip
    LDR     R4, 0x30
    ADDS    R4, R5
    STR     R4, [R0, #UA_THR]

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 22

### 3.4.3 Case2 Приймання даних з ПК

Для реалізації приймання даних через модуль UART необхідно опитувати регістр статусу буферів UA\_FSR чи не були отримані дані. Якщо біт RX\_EMPTY дорівнює 1, то це значить що буфер пустий, у протилежному разі слід виконати зчитування значення регістра UA\_RBR. У разі якщо у буфері знаходять ще отримані дані, то зчитування необхідно виконувати то тих під допоки RX\_EMPTY знову не стане рівним 1.

```
;Check_FLAG_RX_EMPTY
LDR R0, =UART0_BA
LDR R2, =0x00004000 ; маска для біта RX_EMPTY
LDR R3, [R0, #UA_FSR]
ANDS R3, R2
CMP R3, R2
BEQ RX_empty ;якщо RX_EMPTY=1, то дані не отримувалися
LDR R4, [R0, #UA_RBR] ;зчитуємо прийнятий байт
STR R4, [R0, #UA_THR] ;відправляємо назад
RX_empty
movwf PORTB
return
```

### 3.4.4 Case3 Переривання

Програма обробки переривання може бути доволі простою:

```
LDR R0, =UART0_BA
LDR R4, [R0, #UA_RBR] ;зчитуємо прийнятий байт
STR R4, [R0, #UA_THR] ;відправляємо назад
```

Проте слід додатково подбати про контекстне збереження робочих регістрів, попередньо зберігши їх в стек, а після оброблення переривання їх відновивши.

### 3.4.5 Case4 Матрична клавіатура + UART

Використовуючи результати попередньої лабораторної роботи реалізувати виведення у термінал комп'ютера номер натиснутої кнопки з матричної клавіатури.

#### Контрольні запитання до лабораторної роботи № 3

1. Поясніть призначення регістра **GPB\_MFR**.
2. Поясніть всі необхідні операції ініціалізації модуля UART
3. Виконайте ініціалізацію модуля UART1 .

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 23

4. Наведіть приклад ініціалізації модуля UART0 з тактуванням від внутрішнього тактового генератора 22 МГц.
5. Які налаштування визначаються регістром **UA\_LCR**?
6. Поясніть навіщо перевіряти біти **RX\_empty** та **TX\_EMPTY** під час роботи з модулем UART.
7. Приведіть приклад реалізації затримки на асемблері.
8. Поясніть різницю між командами та **BEQ** та **BNQ**
9. Виконайте ініціалізацію модуля UART2.
10. За допомогою яких інструкцій можливо зберігати в стек та відновлювати дані зі стеку.
11. Поясніть роботу зі стеком у разі виклику декількох вкладених переривань.
12. Поясніть правила стандарту AAPCS (Procedure Call Standart for ARM Arhitecture), зокрема розміщення даних в регістрах r0-r3 у разі виклику підпрограм.

### Література до лабораторної роботи № 3

1. Yiu J. The Definitive Guide to the ARM Cortex-M0 / J. Yiu . – Newnes, 2011. – 518 р. – ISBN: 978-0-12-385477-3. [pp. 43-71]
2. NUC140 Datasheet EN V3.02 [Електронний документ]. – Режим доступу : <http://www.nuvoton.com/resource-files/DA00-NUC140ENF1.pdf> – Назва з екрану. [pp. 54-88, 143-144, 328-366].
3. Вбудовані системи <http://radar.kpi.ua/downloads/>

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 24

## Лабораторна робота № 4

### КЕРУВАННЯ ДРАЙВЕРАМИ ТА ПРИВОДАМИ

#### 4.1 Мета та основні завдання роботи

Отримати навички програмування вбудованих систем для керування драйверами приводів, зокрема керування кроковим двигуном.

#### 4.2 Завдання і порядок виконання роботи

В лабораторній роботі слід створити, побудувати та виконати тестування проекту, що дасть змогу вбудованій системі керувати кроковим двигуном. Завдання доцільно виконувати у такій послідовності:

- Створити новий проект на основі лабораторної роботи № 5. У проекті мають бути представлені файли **SystemInit.s**, **main.s** та файл з адресами регістрів **reg\_def.s** та реалізована можливість сканування клавіатури та взаємодії з ПК через модуль UART. У файлі SystemInit.s має бути наявна ініціалізація тактового генератора, системного таймера SysTick, та портів вводу/виводу, що приєднані до світлодіодів та клавіатури, модуля UART. У файлі main.s має бути наявна програма формування затримки.

- Виконати ініціалізацію GPIO для керування кроковим двигуном.
- У режимі Debugger визначити відповідність сигналів керування, що формуються на навчальній платі до входів драйвера (підр. 5.4).

- Сформувати сигнали керування двигуном в напівкроковому режимі.
- Реалізувати повно кроковий режим керування двигуном.
- Реалізувати реверсний рух двигуна на задану кількість кроків.
- Створити інтерфейс керування двигуном через кнопки клавіатури та термінал ПК.

#### 4.3 Основні теоретичні відомості

##### 4.3.1 Різновиди крокових двигунів

Крокові двигуни вже давно і успішно зарекомендували себе в найрізноманітніших пристроях. Їх можна зустріти в приводах CD-ROM, DVD плеєрах, принтерах, верстатах з ЧПК, системах наведення супутникових антен, в фазообертачах, а також в різноманітному промисловому і спеціальному обладнанні.

Кроковий двигун являє собою безколекторний електродвигун постійного струму з декількома обмотками, в якому струм, що подається в одну з обмоток статора,



Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 25

спричиняє фіксацію ротора. Послідовна активація обмоток двигуна спричинює дискретні кутові переміщення (кроки) ротора.

Порівняно із звичайними двигунами постійного струму, крокові двигуни потребують значно складніших схем керування, які повинні виконувати всі комутації обмоток під час роботи двигуна. Для формування необхідної послідовності імпульсів застосовуються мікроконтролери у зв'язці з силовими елементами - драйверами.

Одним з головних переваг крокових двигунів є можливість здійснювати точне позиціонування і регулювання швидкості без сенсора зворотного зв'язку.

У лабораторному стенді використано кроковий електродвигун системи позиціонування лазерної головки від CD-ROM DRIVE SONY CRX310EE. Це біполярний двигун з чотирма виводами і двома обмотками. Біполярний двигун має одну обмотку в кожній фазі, яка для зміни напрямку магнітного поля повинна переполюсовуватися драйвером. Для такого типу двигуна необхідно мостовий драйвер, або півмостовий з двохполярним живленням. Всього біполярний двигун має дві обмотки і, відповідно, чотири виводи.

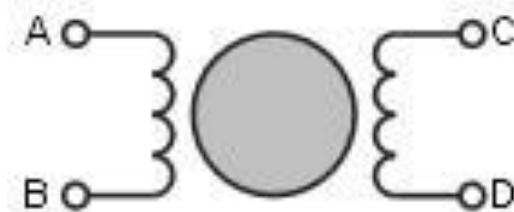


Рис. 4.1 – Схема біполярного крокового двигуна

#### 4.3.2 Драйвер крокового двигуна

Для керування двома обмотками електродвигуна обрано простий чотирьох канальний драйвер для керування електромоторами L293D. Драйвер L293D має чотири входи для керуючих сигналів і чотири виходи для підключення одного крокового двигуна або двох електродвигунів постійного струму. L293D має можливість використовувати роздільне живлення для мікросхеми і для керованих нею двигунів, що дозволяє підключати електродвигуни з напругою, що є більшою ніж у мікросхеми. Роздільне живлення мікроконтролера і електродвигуна також необхідне для зменшення завад, що виникають під час роботи двигунів.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 26

Табл. 4.1 – Характеристики мікросхеми L293D

Характеристика	Значення
напруга живлення двигунів ( $V_s$ ) -	4,5...36 В
напруга живлення мікросхеми ( $V_{ss}$ ) -	5 В
допустимий струм в навантаженні (на кожен канал)	600 мА
піковий (максимальний) струм на виході (на кожен канал)	1,2 А
лог. "0", не більше	1,5 В
лог. "1"	2,3...7 В
максимальна швидкість перемикання	5 кГц.
захист від перегрівання	Є

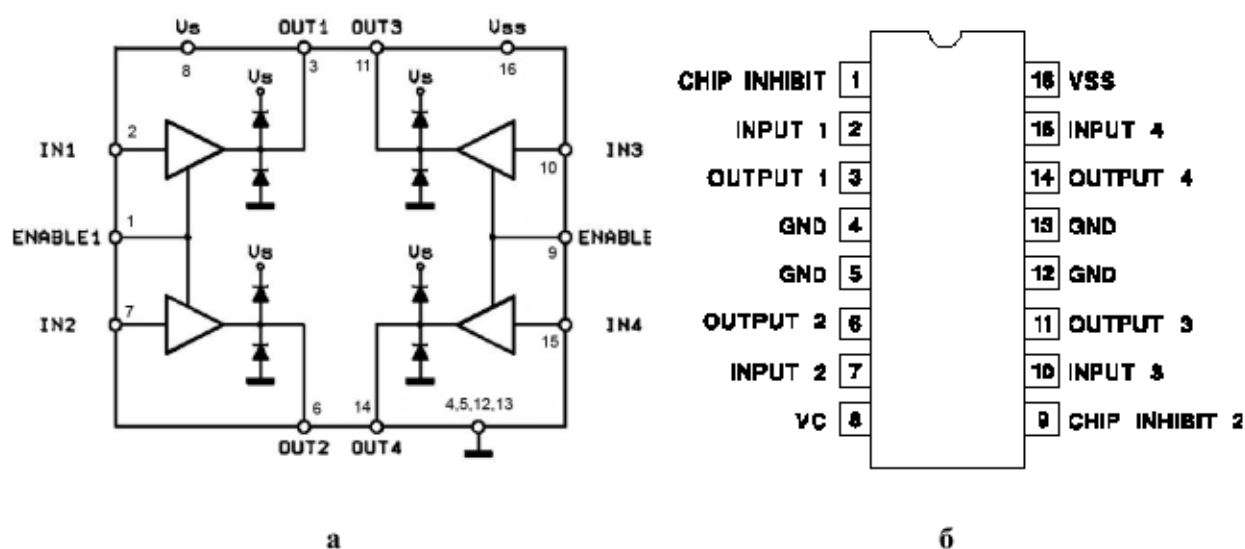


Рис. 4.3 – Структурна схема та виводи IC L293D

На рис. 4.2 використанні наступні позначення:

- ENABLE1 та ENABLE2 – сигнали дозволу каналів 1, 2 та 3, 4.
- INPUT1 и INPUT2 – сигнали керування виходами драйвера OUTPUT1 і OUTPUT2.
- INPUT3 и INPUT4 INPUT2 – сигнали керування виходами драйвера OUTPUT3 и OUTPUT4.
- $V_s$  – напруга живлення для електродвигунів.
- $V_{ss}$  – напруга живлення для мікросхеми.
- GND – "земля".

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 27

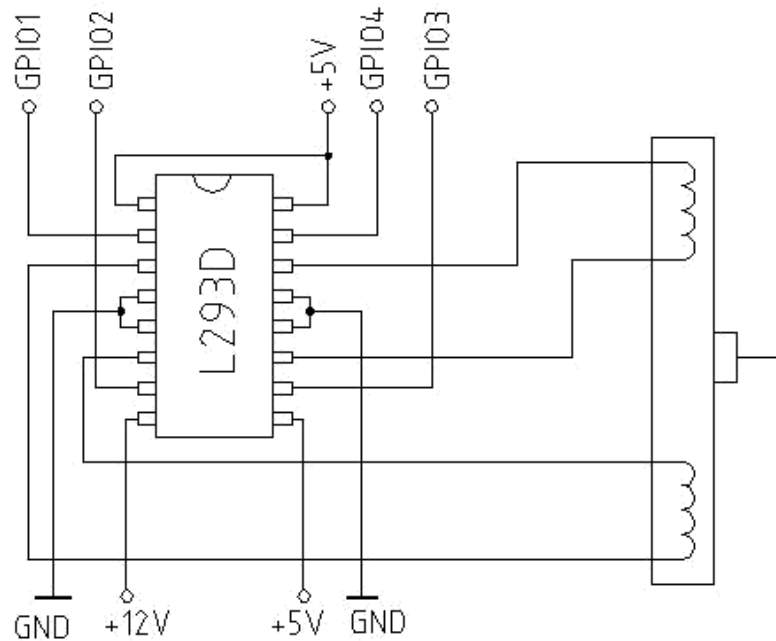


Рис. 4.3 – Схема ввімкнення драйвера L293D

### 4.3.3 Сигнали керування кроковим двигуном

Існує декілька способів керування фазами крокової двигуна. Перший спосіб забезпечується попереминою комутацією фаз, при цьому вони не перекриваються, в один момент часу увімкнена тільки одна фаза (рис. 4.4). Цей спосіб називають "one phase on" full step або wave drive mode. Точки рівноваги ротора для кожного кроку збігаються з «природними» точками рівноваги ротора двигуна без живлення.

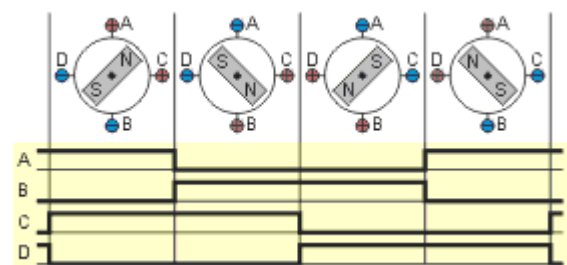
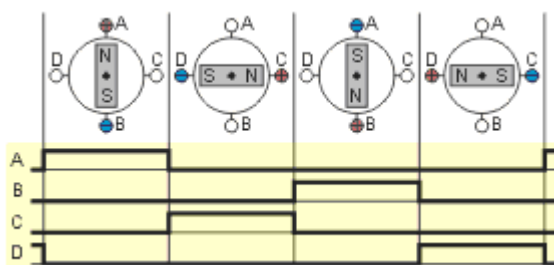


Рис. 4.4 – Повнокроковий режим, Рис. 4.5 – Повнокроковий режим,

ввімкнута одна фаза

ввімкнута дві фази

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 28

Недоліком цього способу керування є те, що для біполярного двигуна в один і той же момент часу використовується 50% обмоток. Це означає, що в такому режимі не може бути отримано повний момент.

Другий спосіб - керування фазами з перекриттям: дві фази включені в один і той же час. Його називають "two-phase-on" full step або просто full step mode. У цьому способі керування ротор фіксується в проміжних позиціях між полюсами статора (рис. 4.5) і забезпечується приблизно на 40% більший момент, ніж у випадку однієї включеної фази. Цей спосіб керування забезпечує такий же кут кроку, як і перший спосіб, але положення точок рівноваги ротора зміщений на пів-кроку.

Третій спосіб є комбінацією перших двох і називається напівкроковим режимом, "one and two-phase-on" half step або просто half step mode, у цьому разі двигун робить крок в половину основного. Цей метод керування досить поширений, оскільки двигун з меншим кроком коштує дорожче і дуже пригодно отримати від 100-крокового двигуна 200 кроків за оберт. Кожен другий крок ввім лише одна фаза, а в інших випадках ввімкнені дві (рис. 4.6).

В результаті кутове переміщення ротора становить половину кута кроку для перших двох способів керування. Окрім зменшення розміру кроку цей спосіб керування дозволяє частково позбутися явища резонансу.

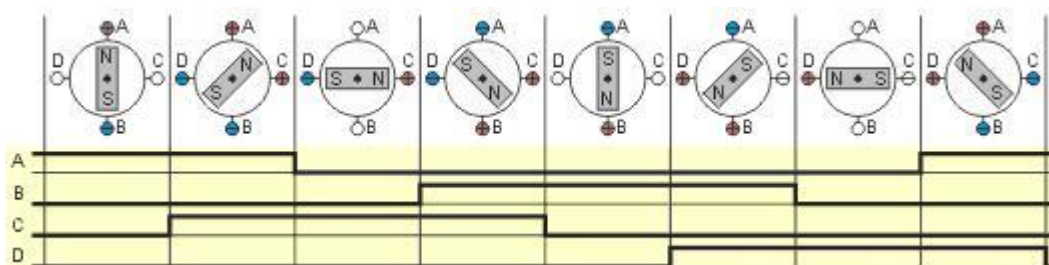


Рис. 4.6 – Напівкроковий режим, ввімкнута одна фаза

Напівкроковий режим зазвичай не дозволяє отримати повний момент, хоча найбільш досконалі драйвери реалізують модифікований напівкроковий режим, в якому двигун забезпечує практично повний момент, при цьому потужність розсіювання не перевищує номінальної.

Ще один спосіб керування називається мікрокроковим режимом або micro stepping mode. У цьому способі керування струм в фазах потрібно міняти невеликими кроками, забезпечуючи таким чином дроблення половинного кроку на ще менші мікрокроки.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 29

Коли одночасно включені дві фази, але їх струми не рівні, то положення рівноваги ротора буде знаходитись не в середині кроку, а в іншому місці, визначеному співвідношенням струмів фаз. Міняючи це співвідношення, можна забезпечити деяку кількість мікрокроків всередині одного кроку. Для реалізації мікрокрокового режиму потрібні значно складніші драйвери, що дозволяють задавати струм в обмотках з необхідною дискретністю. Напівкроковий режим є окремим випадком мікрокрокового, але він не вимагає формування ступінчастого струму живлення котушок, тому часто реалізується.

В повнокроковому режимі з двома включеними фазами положення точок рівноваги ротора зміщені на пів-кроку. Потрібно відзначити, що ці положення ротор приймає під час роботи двигуна, але положення ротора не може зберігатися незмінним після вимкнення струму обмоток. Тому під час увімкнення або вимкнення живлення двигуна ротор зміщуватиметься на пів кроку. Для того, щоб він не зміщувався під час зупинки, необхідно подавати в обмотки струм утримання. Те ж стосується і для напівкрокового і мікрокрокового режимів. Слід зазначити, що якщо в вимкненому стані ротор двигуна повертався, то під час включення живлення можливе зміщення ротора і на більше, ніж половину кроку .

Струм утримання може бути меншим номінального, оскільки від двигуна з нерухомим ротором зазвичай не потрібно великого моменту. Однак є застосування, коли в зупиненому стані двигун має забезпечувати повний момент, що для крокового двигуна можливо. Така властивість крокового двигуна дозволяє в деяких ситуаціях обходитися без механічних гальмівних систем. Оскільки сучасні драйвери дозволяють регулювати струм живлення обмоток двигуна, формування необхідного струму утримання зазвичай не представляє проблем. Завдання зазвичай полягає просто у відповідній програмній підтримці для керуючого мікроконтролера.

## 4.4 Навчальні приклади

### 4.4.1 Case0 Лічильник для формування сигналів керування в повнокроковому режимі

Для того, щоб сформувати сигнали керування для крокового двигуна необхідно використати додатковий лічильник за яким ідентифікувати номер кроку. Приклад програми такого лічильника наведений нижче.

`count`

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 30

```
MOVS R4, #4
```

```
Blinky_loop
```

```
...
SUBS    R4, #1
CMP     R4, #0
BEQ     count
B       Blinky_loop
```

#### 4.4.2 Case1 Формування сигналів керування двигуна в напівкроковому режимі з одною ввімкненою фазою

Залежно від значення лічильника R4 послідовно формується імпульс в розрядах порта A (у разі якщо саме до нього приєднано драйвер двигуна).

```

LDR     R2, =GPIOA_DOUT ; Step Motor
CMP     R4, #4
BNE     subcount2
MOVS   R3, #8
STR     R3, [R1, R2]
subcount2
CMP     R4, #3
BNE     subcount1
MOVS   R3, #1
STR     R3, [R1, R2]
subcount1
CMP     R4, #2
BNE     subcount0
MOVS   R3, #4
STR     R3, [R1, R2]
subcount0
CMP     R4, #1
BNE     subcount
MOVS   R3, #2
STR     R3, [R1, R2]
subcount
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 31

#### 4.4.3 Case2 Напівкроковий режим роботи

Формування сигналів для напівкрокового режиму подібне, але необхідно збільшити модуль лічби лічильника у два рази і прописати всі можливі значення сигналів керування.

#### 4.4.4 Case3 Інтерфейс керування кроковим двигуном

Інтерфейс керування двигуном можливо реалізувати за допомогою клавіатури або приймаючи команди керування з терміналу ПК. Для цього знадобляться напрацювання попередніх лабораторних робіт.

Можливий варіант реалізації інтерфейсу наведений на рис. 4.7.

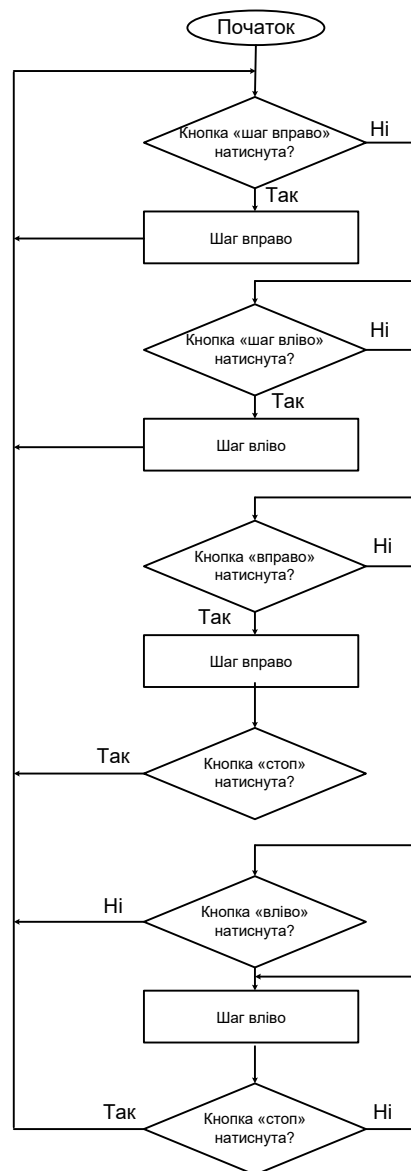


Рис. 4.7 – Інтерфейс керування кроковим двигуном

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 32

### Контрольні запитання до лабораторної роботи № 4

1. Поясніть які необхідно формувати сигнали в напівкроковому режимі керування кроковим двигуном.
2. Які є можливості зменшити крок позиціонування крокового двигуна.
3. Який розмір та діапазон адрес ОЗП в NUC140VE3CN.
4. Який розмір та діапазон адрес Flash пам'яті в NUC140VE3CN.
5. Наведіть приклади циклів for та while на асемблері.
6. Яке максимальне значення константи можливо записати в робочий регістр за допомогою команди **MOVS**.
7. Яке максимальне значення константи можливо додати до робочого регістру за допомогою команди **ADDS**.
8. Поясніть призначення інструкцій **PUSH {R0-R7, LR}** та **POP{R0-R7, PC}**.
9. Як відрізняються між собою інструкції **ADDS** та **ADCS**.
10. Поясніть різницю між командами та **TST** та **CMR**.
11. Як відрізняються між собою інструкції **CMR** та **CMN**.
12. Як збільшити або зменшити швидкість обертання крокового двигуна.

### Література до лабораторної роботи № 4

1. Yiu J. The Definitive Guide to the ARM Cortex-M0 / J. Yiu . – Newnes, 2011. – 518 р. – ISBN: 978-0-12-385477-3. [pp. 43-71]
2. NUC140 Datasheet EN V3.02 [Електронний документ]. – Режим доступу : <http://www.nuvoton.com/resource-files/DA00-NUC140ENF1.pdf> – Назва з екрану. [pp. 54-88, 143-144, 328-366].
3. L293, L293D QUADRUPLE HALF-H DRIVERS: Datasheet [Електронний документ]. – Режим доступу: <http://www.ti.com/lit/ds/symlink/l293d.pdf>.
4. Вбудовані системи <http://radar.kpi.ua/downloads/>



Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 33

## Лабораторна робота № 5

### СТВОРЕННЯ ПЕРШОГО ПРОЕКТУ НА МОВІ C

#### 5.1 Мета та основні завдання роботи

Ознайомитись з особливостями розроблення програмного забезпечення для мікроконтролерів ARM архітектури Cortex-M0 з використанням компілятора середовища «Keil MDK» (ARM RealView Development Suit). Повторити у проектах на мові C основні завдання попередніх лабораторних робіт, зокрема індикацію, сканування клавіатури, вивід даних у термінал комп'ютера.

#### 5.2 Завдання і порядок виконання роботи

В лабораторній роботі слід створити, побудувати та виконати тестування проекту на C. Завдання доцільно виконувати у такій послідовності:

Ознайомитись з теоретичними відомостями

- Запустити майстер проекту для створення проекту та обрати мікроконтролер – NUC140VE3CN;

Створити програму керування індикацією на світло діодах (case1);

Виконати ініціалізацію модуля UART (case2) та перевірити обмін даними між HyperTerminal на ПК та МК на швидкості 115200 бод.

Реалізувати сканування клавіатури з виводом номера натисненої кнопки у термінал ПК.

Налаштувати зовнішнє переривання EINT0 та у разі прийому даних UART02 та виконати оброблення цих переривань, відповідно до case4.

#### 5.3 Основні теоретичні відомості

##### 5.3.1 Порівняння особливостей створення ПЗ для вбудовування систем на асемблері та на C

Процесор Cortex-M0 можна програмувати як на C, так і на асемблері. Також можливо створювати проекти з використанням обох мов програмування одночасно. Початківцям або у разі складних проектів, рекомендується використовувати мову C, оскільки сучасні компілятори доволі ефективно транслюють програми з C на асемблер для мікроконтролерів Cortex.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 34

С	<p>Переваги</p> <ul style="list-style-type: none"> <li>• Легко у вивчені.</li> <li>• Легше перенесення програм з одного процесора на інший</li> <li>• Легша робота з складними структурами даних</li> </ul>	<p>Недоліки</p> <ul style="list-style-type: none"> <li>• Обмежений/відсутній прямий доступ до реєстрів ядра МК та стеку.</li> <li>• Відсутній прямий контроль генерування асемблерних інструкцій</li> <li>• Відсутній прямий контроль використання стеку</li> </ul>
Асемблер	<p>Переваги</p> <ul style="list-style-type: none"> <li>• Прямий контроль над кожним кроком програми та операціями доступу до пам'яті.</li> <li>• Прямий доступ до інструкцій, що не можуть генеруватися на С</li> </ul>	<p>Недоліки</p> <ul style="list-style-type: none"> <li>• Вимагає більше часу для вивчення</li> <li>• Складно оперувати з структурами даних</li> <li>• Складніших перехід на інші процесори (синтаксис асемблера специфічний до процесорів та середовищ розробки програм до МК)</li> </ul>

### 5.3.2 Типи даних

У мові С підтримується численна кількість стандартних типів даних. Проте перелік типів даних може залежати від архітектури процесора та компілятора. У табл. 5.1 приведено типи даних які підтримуються у разі роботи з ARM процесорами, у тому числі Cortex-M0.

Табл. 5.1 – Стандартні типи даних, які підтримують С компілятори

Типи даних С та C99 (stdint.h)	Кількість розрядів	Діапазон (з знаком)	Діапазон (без знаку)
char, int8_t, uint8_t	8	-128 – +127	0–255
short int16_t, uint16_t	16	-32768 – +32767	0 – 65535
int, int32_t, uint32_t	32	-2147483648 – +2147483647	0 – 4294967295
long	32	-2147483648 – +2147483647	0 – 4294967295
long long, int64_t, uint64_t	64	$-2^{63} - +2^{63}-1$	$0-2^{64}-1$
float	32	$-3.4028234 \cdot 10^{38} - +3.4028234 \cdot 10^{38}$	
double	64	$-1.7976931348623157 \cdot 10^{308} - +1.7976931348623157 \cdot 10^{308}$	
long double	64	$-1.7976931348623157 \cdot 10^{308} - +1.7976931348623157 \cdot 10^{308}$	

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 35

pointers	32	0x0 – 0xFFFFFFFF
enum	8/16/32	Найменший можливий тип даних, за виключенням пере визначення опціями компілятора
bool (C++), _Bool (C)	8	True або False
wchar_t	16	0 - 65535

Перелік типів даних у табл. 5.1 підтримуються всіма C компіляторами для ARM процесорів. У табл. 5.2. розглянуті типи даних якими оперують в технічній документації на ARM процесори.

Табл. 5.2 – Визначення типів даних, якими оперують під час роботи з ARM процесорами

Тип даних	Розмірність
Byte	8-bit
Half word	16-bit
Word	32-bit
Double word	64-bit

### 5.3.3 Структури даних та робота з бітами

Окрім змінних в програмах на C необхідно реалізувати доступ до периферійних пристроїв. В мікроконтролерах ARM Cortex-M0, доступ до регістрів периферійних пристроїв виконується за визначеними наперед адресами в пам'яті. Тому для доступу до них, подібно до того як це виконувалось на асемблері, адреси можна визначити через define.

**Увага**, приклади з п. 5.3.3 наведені для кращого розуміння роботи з структурами, повторювати їх необов'язково.

```
#define UA_BASE          0x40050000 // Базова адреса для UART0
#define UA_RBR           (*(volatile unsigned long *) (UA_BASE + 0x00))
#define UA_TRH           (*(volatile unsigned long *) (UA_BASE + 0x00))
#define UA_EIR           (*(volatile unsigned long *) (UA_BASE + 0x04))
#define UA_FCR           (*(volatile unsigned long *) (UA_BASE + 0x08))
#define UA_LCR           (*(volatile unsigned long *) (UA_BASE + 0x0C))
#define UA_MCR           (*(volatile unsigned long *) (UA_BASE + 0x10))
#define UA_MSR           (*(volatile unsigned long *) (UA_BASE + 0x14))
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 36

```
#define UA_FSR          (*(volatile unsigned long *) (UA_BASE + 0x18))
#define UA_ISR          (*(volatile unsigned long *) (UA_BASE + 0x1C))
#define UA_TOR          (*(volatile unsigned long *) (UA_BASE + 0x20))
#define UA_BAUD         (*(volatile unsigned long *) (UA_BASE + 0x24))
#define UA_IRCR         (*(volatile unsigned long *) (UA_BASE + 0x28))
#define UA_ALT_CSR     (*(volatile unsigned long *) (UA_BASE + 0x2C))
#define UA_FUN_SEL     (*(volatile unsigned long *) (UA_BASE +
                                0x30))
```

```
/* ----- UART Initialization ----- */
```

```
void uartinit(void) //Ініціалізація модуля
```

```
UART {
```

```
    UA_FUN_SEL = 0 ; // UART
```

```
    UA_BAUD = 0x3f000066; // DIV_X_EN=1 ; DIV_X_ONE=1 ; A=0x66
```

```
    UA_TOR = (1 << 12); // TX Delay Time Value
```

```
    UA_LCR = 0x3; // 1 StopBits; No Parity
```

```
    UA_FCR = 0x6; // Set Rx Trigger Level (FCR.RFITL)
```

```
    UA_RSR = 0xA; // Clear buffer overrun if any
```

```
}
```

```
/* ----- Transmit a character ----- */
```

```
int sendchar(int ch)
```

```
{
```

```
    while (UA_FSR & 0x10000000); // Busy,
```

```
    wait UA_THR = ch; // write character
```

```
    return ch;
```

```
}
```

```
/* ----- Receive a character ----- */
```

```
int getkey(void)
```

```
{
```

```
    while ((UA_FSR & 0x4000)==0); // No data,
```

```
    wait return UA_RBR; // read character
```

```
}
```

Слід зазначити, що у разі якщо визначається константа як вказівник на адресу в пам'яті, то необхідно додатково використовувати ключове слово “*volatile*”. Як тільки назви регістрів будуть визначені їх можна використовувати у програмі. Для простих

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 37

програм такий підхід цілком може влаштувати, проте якщо у мікроконтролера є три модулі UART, то кожен доведеться визначати окремо. Крім того, визначаючи кожен регістр через константу при, неодмінно призведе до збільшення обсягу пам'яті програми, оскільки кожне визначення регістрів буде збереженим як 32-розрядна константа. Для спрощення програми можна набір регістрів, що стосуються певного периферійного пристрою визначити через структуру даних. Далі приведено приклад визначення через структуру модуля UART. Доступ до регістрів структури виконується через вказівник на структуру.

```

typedef struct { // Base on ARM NUC1xx
    volatile unsigned long RBR;           // 0x00
    volatile unsigned long TRH;           // 0x00
    volatile unsigned long EIR;           // 0x04
    volatile unsigned long FCR;           // 0x08
    volatile unsigned long LCR;           // 0x0C
    volatile unsigned long MCR;           // 0x10
    volatile unsigned long MSR;           // 0x14
    volatile unsigned long FSR;           // 0x18
    volatile unsigned long ISR;           // 0x1C
    volatile unsigned long TOR;           // 0x20
    volatile unsigned long BAUD;          // 0x24
    volatile unsigned long IRCR;          // 0x28
    volatile unsigned long ALT_CSR;       // 0x2C
    volatile unsigned long FUN_SEL;       // 0x30
} UART_TypeDef;

#define Uart0 (( UART_TypeDef *) 0x40050000)
#define Uart1 (( UART_TypeDef *) 0x40150000)
#define Uart2 (( UART_TypeDef *) 0x40154000)

/* ----- UART Initialization ----- */

void uartinit(void) // Simple initialization for Primecell PL011
{
    Uart0->FUN_SEL = 0 ;                // UART
    Uart0->BAUD = 0x3f000066;           // DIV_X_EN=1 ; DIV_X_ONE=1 ; A=0x66
    Uart0->TOR = (1 << 12);             // TX Delay Time Value

    Uart0->LCR = 0x3;                   // 1 StopBits; No Parity

```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 38

```
Uart0->FCR = 0x6;          // Set Rx Trigger Level (FCR.RFITL)
Uart0->RSR = 0xA;         // Clear buffer overrun if any
}
```

```
/* ----- Transmit a character ----- */
```

```
int sendchar(int ch)
```

```
{
    while (Uart0->FSR & 0x10000000); // Busy, wait
    Uart0->THR = ch; // write character return ch;
}
```

```
/* ----- Receive a character ----- */
```

```
int getkey(void)
```

```
{
    while ((Uart0->FSR & 0x4000)==0); // No data, wait
    return Uart0->RBR; // read character
}
```

У вище приведеному прикладі для доступу до регістру Integer Baud Rate Divider (IBRD) модуля UART#0 слід користуватися виразом Uart0-> IBRD, а ; разі якщо необхідно отримати доступ до такого ж регістра модуля UART#1 – Uart1-> IBRD.

Для того, щоб зробити універсальними ще і функції, що використовуються для роботи з периферійним пристроєм слід їх слід дещо модифікувати. Зокрема у функцію слід передавати вказівник на модуль як параметр:

```
/* ----- UART Initialization ----- */
```

```
void uartinit(UART_Typedef *uartptr) //
```

```
{
    uartptr->FUN_SEL = 0 ;          // UART
    uartptr->BAUD = 0x3f000066;     // DIV_X_EN=1;DIV_X_ONE=1; A=0x66
    uartptr->TOR = (1 << 12);      // TX Delay Time Value
    uartptr->LCR = 0x3;            // 1 StopBits; No Parity
    uartptr->FCR = 0x6;           // Set Rx Trigger Level (FCR.RFITL)
    uartptr->RSR = 0xA;           // Clear buffer overrun if any
}
```

```
/* ----- Transmit a character ----- */
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 39

```
int sendchar(UART_Typedef *uartptr, int ch)

{

    while (uartptr->FSR & 0x10000000); // Busy, wait
    uartptr->THR = ch; // write character
    return ch;

}

/* ----- Receive a character ----- */

int getkey(UART_Typedef *uartptr)

{

    while ((uartptr->FSR & 0x4000)==0); // No data, wait
    return uartptr->RBR; // read character

}
```

### 5.3.4 Створення проекту

Створення проекту на С в покроковій інструкції не відрізняється від випадку використання асемблера. Необхідно обрати тип процесора та згодитись, щоб до проекту був доданий шаблон файлу загрузки startup\_NUC1xx.s . Відмінність буде полягати у тому, що функції main та SystemInit (викликаються з файлу загрузки) вже будуть реалізовані на С. Далі приведено приклад визначення цих функцій у файлі main.c.

```
#include "NUC1xx.h"
int main (void) {
}
void SystemInit (void) {
}
```

У файлі NUC1xx.h виконується визначення всіх регістрів та бітів налаштувань периферійних пристроїв та ядра процесора через структури

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 40

(більше 6000 стрічок). Даний файл виконаний розробником мікроконтролерів та дозволяє пришвидшити процес розробки програм на С. Він є аналогічним файлу `reg_def.s` яким ми створювали на попередніх лабораторних роботах самостійно.

Файл `NUC1xx.h` входить в бібліотеку `BSP_Library` (встановлюється окремо від `µVision`). Тому у проекті слід вказати шлях до директорії в якій він знаходиться. Крім того у самому `NUC1xx.h` також підключаються файли `core_cm0.h`, `system_NUC1xx.h` та `System\SysInfra.h`. Налаштування виконується в меню за адресою `Project > Options for Target` або (`Alt+F7`) у закладці “C/C++”, поле “Include Path” (рис. 5.1). Шляхи до директорій з файлами, які підключаються в проекті, приведені на рис. 5.2. Опісля виконання цих налаштувань проект на С має успішно компілюватися.

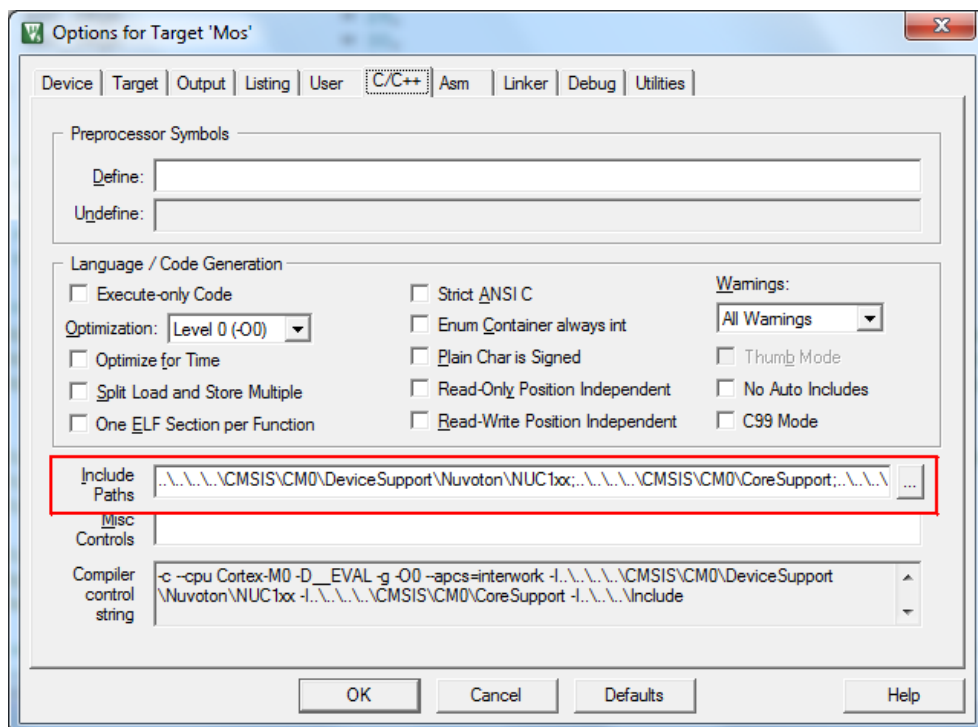


Рис. 5.1 – Налаштування проекту на C/C++



Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 41

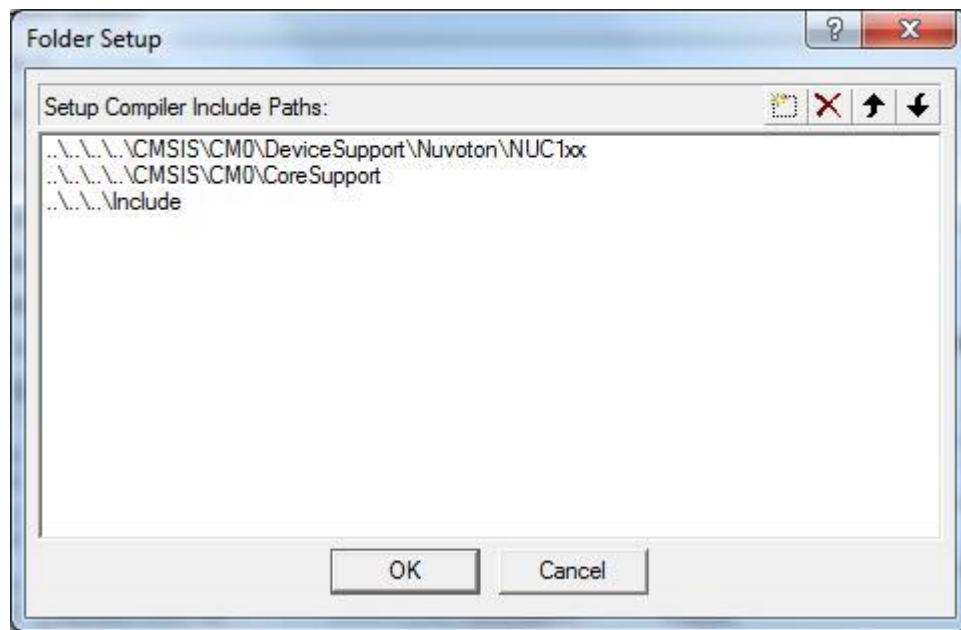


Рис. 5.2 – Відносні адреси з BSP\_Library, що слід додати в проект

## 5.4 Навчальний приклад

### Case1 Blinky Project – Індикація за допомогою світлодіодів

У даному кейсі слід повторити на C програму з першої лабораторної роботи, що була виконана на асемблері. Приклад такої програми приведений нижче.

```
int main (void) {
    int i;
    // SysTick Timer Initialization

    SysTick->LOAD = (48000-1);           // Count from 47999 to 0

    SysTick->VAL = 0;                     // Clear SysTick value
    SysTick->CTRL = 0x5;                   // Enable, using core clock
    //GPIO Init
    GPIOC->PMD.PMD14 = 0x1;
    GPIOC->PMD.PMD15 = 0x1;
    GPIOC->DOUT = 0x1;
    while(1) {
        for (i=0; i<500;i++) { // Wait for 0.5 seconds
            while ((SysTick->CTRL & 0x10000)==0)
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 42

```

; // Wait for counter underflow
}
GPIOC->DOOUT = ~GPIOC->DOOUT;
}
}

```

Зверніть увагу, що звернення до регістрів виконується у доволі зручній та інтуїтивно зрозумілій формі. З назвами груп регістрів, що об'єднані в структури, та з назвами окремих бітових полів можна ознайомитись у файлі `NUC1xx.h`. При цьому зручно користуватись пошуком. Зокрема знайдіть структури, що описують регістри системного таймера **SysTick** та портів вводу виводу загального призначення **GPIOC**, **PMD**.

Також змініть програму для того, щоб на індикацію виводилось значення лічильника та біжуча змійка.

## Контрольні запитання

1. Приведіть приклад розрахунку затримки в 1 мс за допомогою системного таймера *SysTick* у разі якщо тактова частота процесора рівна 48 МГц.
2. Яка різниця між програмою в якій певний стан реалізується через опитування стану периферійного пристрою та програмою з обробкою переривання.
3. Навіщо перед передаванням даних у термінал комп'ютера виконувати перетворення даних у двійково-десятковий та ASCII код.
4. Як саме відрізняється представлення цифр у ASCII коді від їх представлення у двійково-десятковому.
5. Яка перевага використання апаратного перемножувача перед програмною реалізацією цієї процедури.
6. Яке призначення сценарію компонування проекту *Linker*.
7. Чому у дизасембльованому лістингу проста програма на С реалізується значною кількістю команд на асемблері.
8. Якими командами на асемблері можна реалізувати оператори *while* та *for*.
9. Як на С реалізуються операції з бітами?
10. Поясніть механізм роботи з масивом `buff[5]` з використанням вказівників.
11. Яке призначення на С мають символи `%` та `&`.
12. Поясніть як через структури даних оперувати з окремими бітами.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 43

## Література до лабораторної роботи № 5

1. Yiu J. The Definitive Guide to the ARM Cortex-M0 / J. Yiu . – Newnes, 2011. – 518 p. – ISBN: 978-0-12-385477-3. [pp. 43-71, 231-258]
2. NUC140 Datasheet EN V3.02 [Електронний документ]. – Режим доступу: <http://www.nuvoton.com/resource-files/DA00-NUC140ENF1.pdf> – Назва з екрану. [pp. 47-151, 177-199].
3. Вбудовані системи <http://radar.kpi.ua/downloads/>

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 44

## Лабораторна робота № 6

### ДИСКРЕТИЗАЦІЯ ТА АНАЛОГОВО-ЦИФРОВЕ ПЕРЕТВОРЕННЯ СИГНАЛІВ

#### 6.1 Мета та основні завдання роботи

Отримати навички реалізації функцій контролю у вбудованих систем на основі реєстрації аналогових сигналів сенсорів та первинних перетворювачів. Засвоїти теоретичні знання стосовно дискретизації та квантування за рівнем аналогових сигналів, усвідомити спосіб реалізації процесу квантування сигналу за рівнем в модулі АЦП МК.

#### 6.2 Завдання і порядок виконання роботи

- В лабораторній роботі слід створити, побудувати та виконати тестування проекту, що дасть змогу вбудованій системі перетворювати аналогові сигнали в цифрові. Завдання доцільно виконувати у такій послідовності:
  - Створити новий проект на С. У проекті мають бути представлені файли асемблера **SystemInit.s**, та файл з адресами регістрів **reg\_def.s** з ініціалізованими тактовим генератором, системним таймером SysTick, портами вводу/виводу, та модулем UART. Також створити файл main.c
  - Виконати ініціалізацію модуля TMR0 (п. 6.3.2) з довільною частотою дискретизації та визначити підпрограму обробки переривання **TMR0\_IRQHandler** (п. 6.4.1).
  - У режимі Debugger впевнитись у тому, що таймер ввімкнутий та формує переривання у разі переповнення. Для цього зручно встановити на векторі переривання точку зупинки (breakpoint) та скористатись функцією «Run».
  - Перевірити роботу модуля UART, відповідно до ініціалізації в **SystemInit.s**, що виконувалась на лаб. роб 4 та перевірити на макеті вивід даних у термінал комп'ютера з частотою, що задає таймер.
  - Виконати ініціалізацію модуля АЦП (п. 6.3.3) та вивід значень АЦП у термінал комп'ютера (6.4.2).
  - Налаштувати формування переривань по завершенню аналогово-цифрового перетворення (п. 6.4.3).
  - Реалізувати алгоритм перетворення двійкового коду в двійково-десятковий (п. 6.3.4 та п. 6.4.4) з передачею його у термінал комп'ютера.
  - Реалізувати нормування значень АЦП для відображення значень фізичної величини.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 45

- Уточнити у викладача індивідуальне завдання **частоту дискретизації** АЦП.

## 6.3 Основні теоретичні відомості

### 6.3.1 Створення проекту з підтримкою переривань

Лабораторна робота складається з 5-ти кейсів, в яких окремо необхідно виконати ініціалізацію модуля таймера TMR0 з формуванням переривання за переповненням для формування частоти дискретизації; виконати ініціалізацію модуля АЦП, а також реалізувати аналогово-цифрове перетворення в реальному часі з виводом даних у термінал ПК. Оскільки аналогово-цифрове перетворення необхідно виконувати з певною частотою дискретизації, то доцільно використати апаратно реалізований таймер з налаштуванням переривання. Основна програма буде мати наступний вигляд:

```
int main (void) {
    InterruptInit();
    Timer_initial();
    InitADC();
    while(1) ;
}
```

### 6.3.2 Ініціалізація модуля TMR0

Ознайомтесь у документації на МК NUC140 з розділом 10 Timer Controller (TMR), зокрема з регістрами, які напряму чи опосередковано впливають на режими роботи модуля. З'ясуйте для себе у якому режимі має працювати модуль для забезпечення частоти дискретизації. Зробіть відповідні розрахунки для встановлення частоти переривань TMR0 за переповненням в 100 Гц, враховуючи, тактову частоту, що була обрана для модуля таймера.

Для налаштування переривання за переповненням таймера 0 необхідним також буде ознайомлення з контролером переривань МК (NVIC) п. 5.2.3 документації на мікроконтролер NUC140.

```
void Timer_initial(void) {
/* Step 1. Enable and Select Timer clock source */ SYSCLK-
    >CLKSEL1.TMR0_S = 0; //Select 12Mhz for Timer0 clk src SYSCLK-
    >APBCLK.TMR0_EN =1; //Enable Timer0 clock source
/* Step 2. Select Operation mode */ TIMER0->TCSR.MODE=1; //Sel.
    periodic mode for operation mode
/* Step 3. Select Time out period = (Period of timer clock input) *
(8-bit Prescale + 1) * (24-bit TCOMP)*/
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 46

```

TIMER0->TCSR.PRESCALE=0; // Set Prescale [0~255] TIMER0-
>TCMPR = 1000000;// Set TCCR(TCMP) [0~16777215]
// (1/22118400)*(0+1)*(2765)= 125.01usec or 7999.42Hz
/* Step 4. Enable interrupt */
TIMER0->TCSR.IE = 1;

TIMER0->TISR.TIF = 1; //Write 1 to clear for safty

NVIC_EnableIRQ(TMR0_IRQn); //Enable Timer0 Interrupt
/* Step 5. Enable Timer module */
TIMER0->TCSR.CRST = 1; //Reset up counter
TIMER0->TCSR.CEN = 1; //Enable Timer0
TIMER0->TCSR.TDR_EN=1; // Enable TDR function
}

```

### 6.3.3 Ініціалізація модуля АЦП

Ознайомтесь у документації на МК NUC140 з розділом 5.16 Analog-to-Digital Converter (ADC), зокрема з регістрами які напряму чи опосередковано впливають на режими роботи модуля.

З'ясуйте для себе як налаштувати так, щоб АЦП працював з ПВВ до якого приєднаний регулятор - подільників напруги на макеті (рис. 6.1), далі які налаштування необхідно зберегти в регістрах та знайдіть алгоритм перетворення. Зробіть відповідні обрахунки приведені у підр. 5.16, зокрема часової затримки для схеми вибірки зберігання (sample&hold).

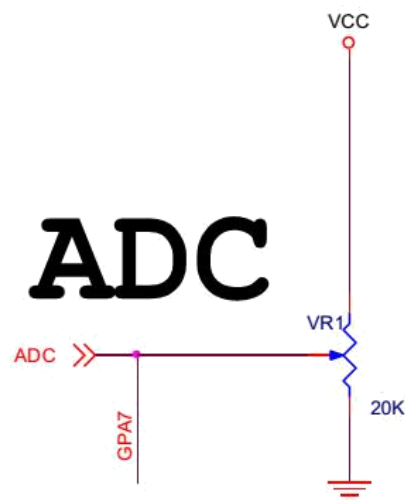


Рис. 6.1 – Подільник напруги, що приєднаний до GPA7

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 47

```

void InitADC(void) {
/* Step 1. GPIO initial */
GPIOA->OFRD|=0x00800000;           //Disable digital input path
SYS->GPAMP.ADC7_SS21_AD6=1;       //Set ADC function
/* Step 2. Enable and Select ADC clock source, and then enable ADC module */
SYSCLK->CLKSEL1.ADC_S = 2;         //Select 22Mhz for ADC
SYSCLK->CLKDIV.ADC_N = 1; //ADC clock source=22Mhz/2=11Mhz;
SYSCLK->APBCLK.ADC_EN = 1;        //Enable clock source
ADC->ADCR.ADEN = 1;                //Enable ADC module
/* Step 3. Select Operation mode */
ADC->ADCR.DIFFEN = 0;              //single end input
ADC->ADCR.ADMO = 0;                //single mode
/* Step 4. Select ADC channel */
ADC->ADCHER.CHEN = 0x80;
/* Step 5. Enable ADC interrupt */
ADC->ADSR.ADF = 1; //clear the A/D interrupt flags for safe
ADC->ADCR.ADIE = 1;

NVIC_EnableIRQ(ADC_IRQn);
/* Step 6. Enable WDT module */
ADC->ADCR.ADST=1;
}

```

### 6.3.4 Case1 Функції на C

Для отримання навичок написання функцій C для мікроконтролерів виконаємо окремими функціями ініціалізацію модулів TMR0, ADC, а також функцій для передачі даних через UART.

Зверніть увагу на формат і стиль реалізації в C роботи з бітами. UART0->FSR.RX\_EMPTY це структура яка розписує регістри модуля UART та біти конкретно в регістрі FSR. Для звернення до певного біта структури використовується крапка і назва конкретного біта. Знайдіть файл NUC1xx.h, який підключений на початку програми та знайдіть у ньому розписані структури на усі регістри мікроконтролера NUC140. Також файл core\_cm0.h у якому, розписані структури на системні регістри.

Виклик функцій ініціалізації виконується наступним чином:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 48

**Timer\_initial() ;                      InitADC() ;**

Для передачі одного байту даних напишемо іншу функцію:

```
void uart_tx_char(unsigned char data)
{
    while(UART0->FSR.TE_FLAG == 0)
        ;
    UART0->DATA = data;
}
```

Перевірте чи відбувається передача даних у термінал комп'ютера. Для цього у нескінченному циклі `while(1)` після лічильника додайте виклик функції **uart\_tx\_char(counter)**. Далі, якщо дані передаються у термінал комп'ютера, то реалізуємо функцію що зможе коректно відображати дані у терміналі. Для цього необхідно реалізувати функцію перетворення з двійкового коду у двійково-десятковий і далі в ASCII. Нижче приведено дві функції **hex\_to\_ascii** та **uart\_transmit**, що реалізують таке перетворення. Також для нових функцій необхідно об'явити прототипи функцій на початку програми, наприклад:

```
void uart_tx_char(unsigned char data);
void hex_to_ascii(unsigned int binary, unsigned char *ascii);
void uart_transmit(void);
void InitUSART(void);

void uart_transmit(unsigned int data)
{
    unsigned char buff[5];
    hex_to_ascii(data, &buff[0]);

    uart_tx_char('V');
    uart_tx_char(buff[4]);
    uart_tx_char(buff[3]);
    uart_tx_char(buff[2]);
    uart_tx_char(buff[1]);
    uart_tx_char(buff[0]);
    uart_tx_char('\n');
    uart_tx_char('\r');
}
```



Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 49

```

void hex_to_ascii(unsigned int binary, unsigned char *ascii)
{
    register unsigned    char n;
    for(n = 0; n <= 4; n++)// очистка
        массиву ascii[n] = '0';
    for(n = 0; n <= 4; n++)//перетворення в двійково-десятковий код
    {
        ascii[n] = (binary % 10) | '0';
        binary = binary / 10;
        if(binary == 0)    break;
    }
}

```

Якщо у циклі `while(1)` замість `uart_tx_char(counter)` викликати функцію `uart_transmit()`, то результат буде такий як приведено на рис. 6.2. Спробуйте побачити як зміниться вивід даних у термінал, якщо змінювати тип даних змінної лічильника `counter` з `unsigned char` на `unsigned int`.

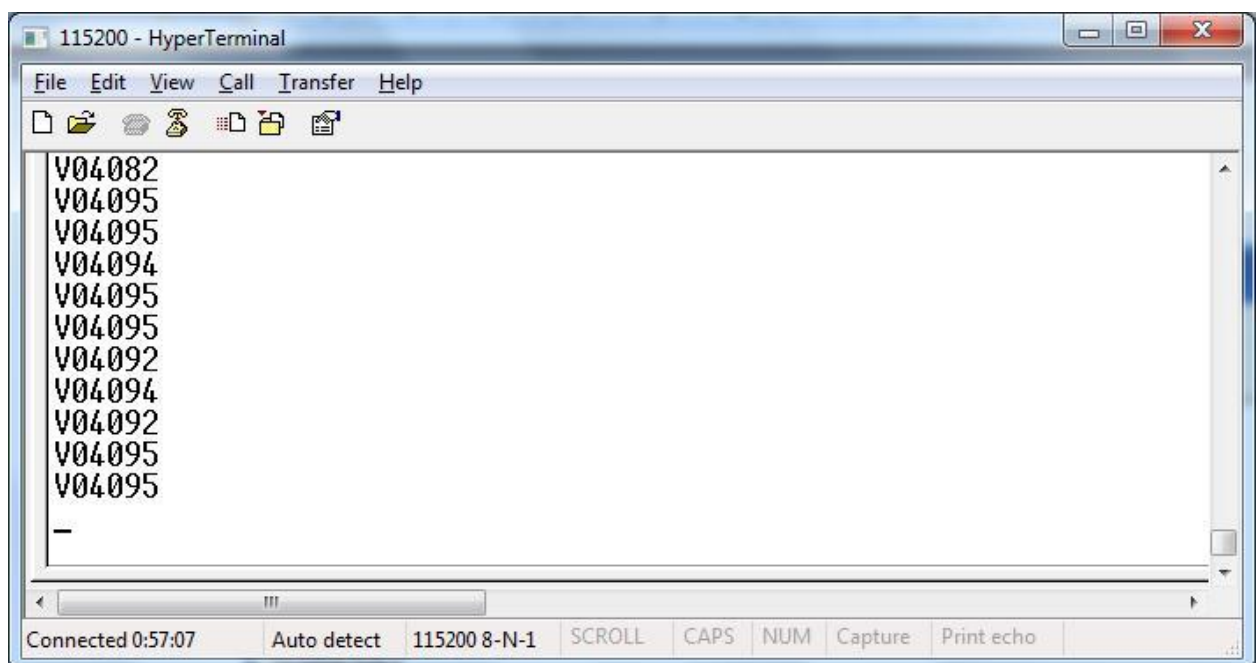


Рис. 6.2 – Вивід у термінал значення 16 розрядного лічильника з використанням перетворення двійкового коду в двійково-десятковий, сумісний з таблицею ASCII

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 50

## 6.4 Навчальні приклади

### 6.4.1 Case1 Оброблення переривання за переповненням TMR0

Програма першого кейсу дозволяє перевірити чи вірно налаштований послідовний порт для передачі даних, а також термінал комп'ютера для прийому даних.

```

unsigned char count;

void TMR0_IRQHandler(void) // Timer0 interrupt subroutine
{
    TIMER0->TISR.TIF =1;

    count++;

    if(count==10)
        count = 0;

    UART0->DATA =  count + 0x30;

    UART0->DATA = '\n';

    UART0->DATA = '\r';

}

```

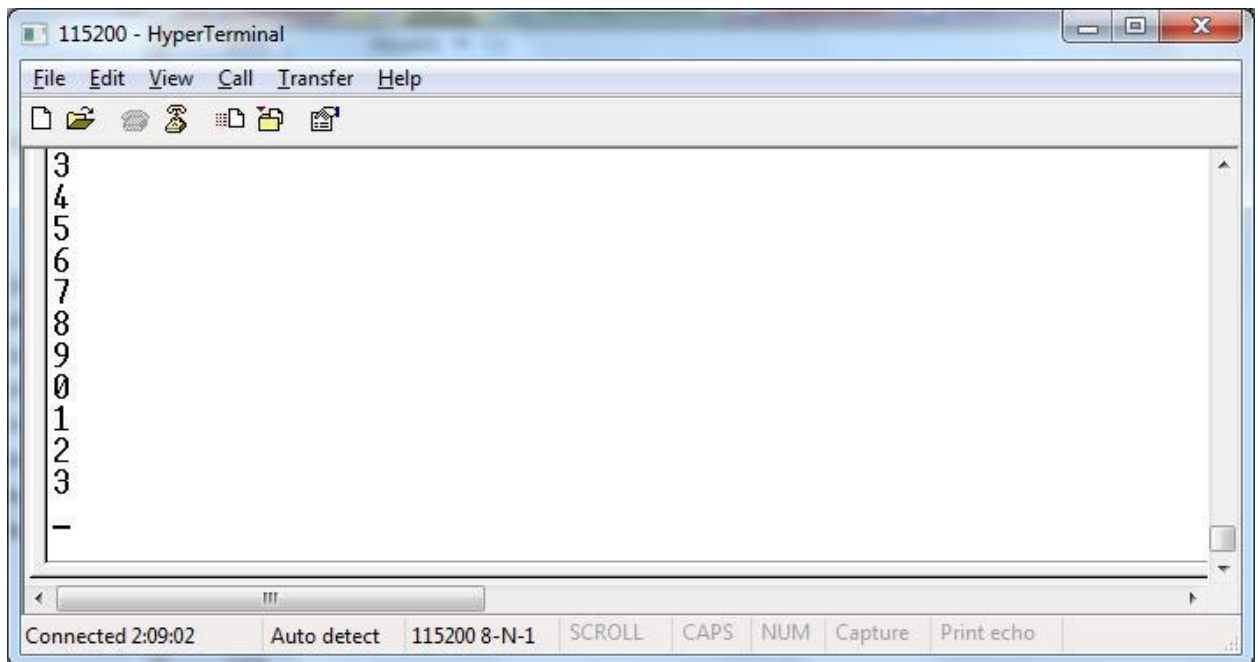


Рис. 6.3 – Передача даних через UART з частотою переривань від TMR0

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 51

### 6.4.2 Case2 Аналогово-цифрове перетворення

Далі слід у підпрограми обробки переривання за переповненням таймера запускати аналогово-цифрове перетворення та зчитувати цифрове значення. При цьому основна програма буде мати такий вигляд:

Результат аналогово-цифрового перетворення є 12-ти розрядним. Для контролю роботи АЦП виведемо молодший байт перетвореного у цифровий код значення аналогової величини у термінал ПК через UART. При цьому ми будемо спостерігати ASCII символи, що відповідають значення АЦП.

```
void TMR0_IRQHandler(void) // Timer0 interrupt subroutine {
    TIMER0->TISR.TIF =1;

    ADC->ADSR.ADF=1;

    //while (ADC->ADSR.ADF==0)
    //      ;

    UART0->DATA =    ADC->ADDR[7].RSLT;

    ADC->ADCR.ADST=1;
}
```

У разі змін положення подільника напруги має змінюватись і значення ASCII символів у терміналі, що характеризують цифрові значення напруги (рис. 6.4).

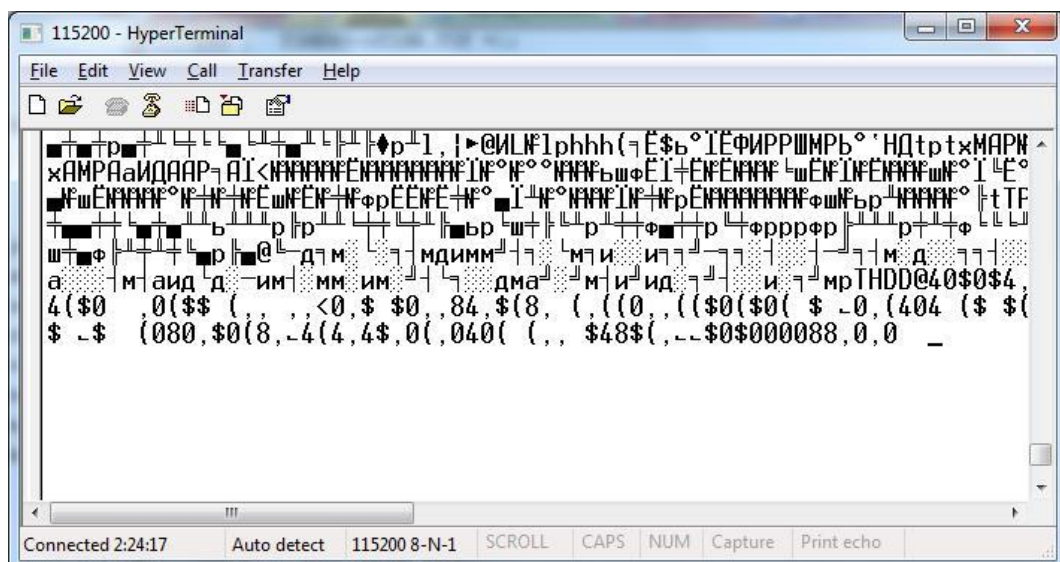


Рис. 6.4 – Передача даних з АЦП через UART у термінал комп'ютера

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 52

Для того, щоб передавати значення зручні для сприйняття людиною, двійковий код з АЦП слід перетворити у двійково-десятковий, а далі в ASCII. Дану операцію можна виконати за допомогою набору функцій, що запускаються через `uart_transmit`, наприклад:

```
uart_transmit(ADC->ADDR[7].RSLT);
```

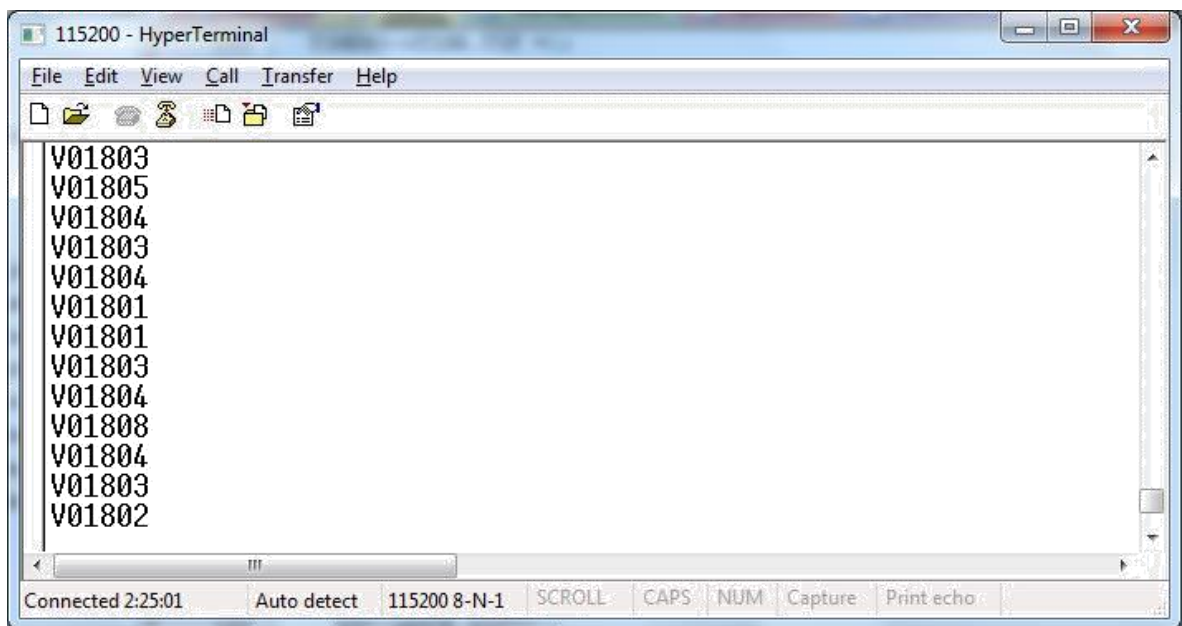


Рис. 6.5 – Передача даних через функцію `uart_transmit()`

### 6.4.3 Case3 Нормування цифрового коду з АЦП

Значення з виходу АЦП необхідно нормувати на  $5V/2^{12}$ , що реалізовано множенням на  $5V/2^{12} * 256$  з наступним діленням результату на 256, тобто зсувом на 8 розрядів вліво.

#### Контрольні запитання

1. Приведіть формулу розрахунку частоти переривань таймера.
2. Навіщо перед передаванням даних у термінал комп'ютера виконувати перетворення даних у двійково-десятковий та ASCII код.
3. Як залежить точність формування частоти дискретизації від вибору значення попереднього подільника таймера.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 53

4. З яких міркувань обирається частота дискретизації.
5. З якою метою слід використовувати переривання від АЦП.
6. Що означають та виконують службові символи в ASCII: «\n», «\r», «\t», «\f».
7. Поясніть алгоритм перетворення двійкового коду в двійково-десятковий на прикладі функції *hex\_to\_ascii()*.
8. Чим відрізняються між собою регістри **NVIC\_IUSER** та **NVIC\_ICER** ?
9. Як встановлюються пріоритети на переривання МК ARM архітектури.
10. Яка різниця між перериваннями та виключеннями в МК ARM архітектури?
11. Що саме виконується наступним налаштуванням `SYS->GPAMFP.ADC7_SS21_AD6=1;`
12. У якому з регістрів налаштування виконується ввімкнення (дозвіл) модуля таймера 0.

#### Література до лабораторної роботи № 6

- 518 Yiu J. The Definitive Guide to the ARM Cortex-M0 / J. Yiu . – Newnes, 2011. – 518 p. – ISBN: 978-0-12-385477-3. [pp. 43-71, 231-258]
2. NUC140 Datasheet EN V3.02 [Електронний документ]. – Режим доступу : <http://www.nuvoton.com/resource-files/DA00-NUC140ENF1.pdf> – Назва з екрану. [pp. 47-151, 177-199].
3. Вбудовані системи <http://radar.kpi.ua/downloads/>

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 54

## Лабораторна робота № 7

### СТВОРЕННЯ УНІФІКОВАНИХ ПРОЕКТІВ НА С ДЛЯ ARM ПРОЦЕСОРІВ

#### 7.1 Мета та основні завдання роботи

Отримати навички реалізації уніфікованих проектів для ARM Cortex-M процесорів з метою пришвидшення розробки програмного забезпечення для вбудованих систем та можливості легкого портування проектів на ARM процесори від інших виробників та з іншою комплектацією та реалізацією периферійних пристроїв.

#### 7.2 Завдання і порядок виконання роботи

Протягом лабораторної роботи слід реалізувати ряд завдань з максимальним використанням готових рішень стосовно ініціалізації периферійних пристроїв, переривань та інтерфейсів взаємодії з зовнішніми пристроями. Завдання доцільно виконувати у такій послідовності:

- На основі попереднього проекту створити новий, у якому функції для виводу інформації у термінал ПК через UART інтерфейс замінити на стандартні функції з бібліотеки **stdio.h**, зокрема **printf()** та **scanf()**.

Порівняти підходи у програмуванні. Виконати кейси стосовно виводу даних та розпізнавання прийому даних за певним шаблоном.

- Змінити ініціалізацію периферійних пристроїв (тактовий генератор, системний таймер SysTick, портами вводу/виводу, модуль UART) з асемблера на С з використанням структур, що прописані в бібліотеці CMSIS. При цьому файл асемблера **SystemInit.s** вилучити з проекту.

- Виконати ініціалізацію модуля формування ШІМ (PWM) та за допомогою подільника напруги та АЦП виконати зміну гучності та частоти зумера на навчальній платі.

- Виконати ініціалізацію SPI інтерфейсу та LCD індикатора з використанням запропонованого у бібліотеці BSP Library драйверу та бібліотеки функцій для виводу даних на LCD індикатор.

- Перевизначити базові функції для стандартної бібліотеки для можливості виводу даних на індикатор через **printf()**.

#### 7.3 Основні теоретичні відомості

##### 7.3.1 Стандарти бібліотеки С

У проектах на С часто використовуються функції з стандартних бібліотек, зокрема стосовно обміну даними через інтерфейси користувача або ж у разі необхідності

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 55

виконувати математичні розрахунки та у інших випадках. Розглянемо функції з бібліотеки `stdio.h`, зокрема функції `printf` та `scanf`.

### Синтаксис функції `printf`.

```
#include <stdio.h>
int printf ( const char *fmtstr , arguments );
```

За допомогою функції **printf** зручно формувати дані (символи та цифрові значення) для виводу у потіці за допомогою функції виводу одного символу **putchar**. Аргумент `fmtstr` визначає формат стрічки у якому інформація буде виведена потоком.

Звичайні символи та символи контролю терміналу копіюються у потік без змін у тому порядку як вони вказані у шаблоні. Визначення елементів форматування завжди починаються з знаку `'%'` і у разі його використання вимагає додаткових аргументів у функції.

Загальний синтаксис елементів форматування: `% flags width .precision {l|L} type`, де `flags` визначає знак або префікс до шістнадцяткового або вісімвоногозначення; `width` – кількість символів на одне значення; `type` – визначає тип аргумента: символ, стрічка, число або вказівник, як приведено у табл. 7.1.

Табл. 7.1 – Службові символи елементів форматування

Символ	Тип аргументу	Вхідний формат
d	int	Знакове десяткове число.
u	unsigned int	Беззнакове десяткове число.
o	unsigned int	Число у 8-ми розрядному форматі.
x	unsigned int	Беззнакове шістнадцяткове число "0123456789abcdef".
X	unsigned int	Беззнакове шістнадцяткове число "0123456789ABCDEF".
f	float	Число з плаваючою комою у форматі <code>&lt;[&gt;-&lt;]&gt;dddd.dddd</code> .
e	float	Число з плаваючою комою у форматі <code>&lt;[&gt;-&lt;]&gt;d.dddde&lt;[&gt;-&lt;]&gt;dd</code> . Число з плаваючою комою у форматі
E	float	<code>&lt;[&gt;-&lt;]&gt;d.ddddE&lt;[&gt;-&lt;]&gt;dd</code> . Число з плаваючою комою у форматі e або f, залежно від того який з
g	float	них буде більш компактним для конкретного значення Число з плаваючою комою у форматі E або f, залежно від того який з
G	float	них буде більш компактним для конкретного значення.
c	char	Один символ.
s	*	Стрічка символів, що закінчується нульовим символом ( <code>'\0'</code> ).
p		Вказівник у форматі адреси

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 56

Приклад:

```
#include <stdio.h>

void tst_printf (void) {
    char a = 1;
    int b = 12365;
    long c = 0x7FFFFFFF;

    unsigned char x = 'A';
    unsigned int y = 54321;
    unsigned long z = 0x4A6F6E00;

    float f = 10.0;
    float g = 22.95;

    char buf [] = "Test String";
    char *p = buf;

    printf ("char %d int %d long %ld\n", a, b, c);
    printf ("Uchar %u Uint %u Ulong %lu\n", x, y, z);
    printf ("xchar %x xint %x xlong %lx\n", x, y, z);
    printf ("String %s is at address %p\n", buf, p);
    printf ("%f != %g\n", f, g);
    printf ("%*f != %*g\n", 8, f, 8, g);
}
```

**Синтаксис функції scanf.**

```
#include <stdio.h>
int scanf ( const char *fmtstr , argument... );
```

За допомогою функції **scanf** можливо отримати дані з вхідного потоку, що формується підпрограмою **getchar**. Вхідні дані зберігаються у змінні, що визначаються додатковими аргументами та у відповідності до формату *fmtstr*. Службові символи для інтерпретації вхідного потоку аналогічні як і у **printf**.

Приклад:

```
#include <stdio.h>

void tst_scanf (void) {
```



Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 57

```

char a;

int b;

long c;

unsigned char x;
unsigned int y;
unsigned long z;

float f,g;
char d, buf [10];
int argsread;

printf ("Enter a signed byte, int, and long\n");
argsread = scanf ("%bd %d %ld", &a, &b, &c);
printf ("%d arguments read\n", argsread);

printf ("Enter an unsigned byte, int, and long\n");
argsread = scanf ("%bu %u %lu", &x, &y, &z); printf
("%d arguments read\n", argsread);

printf ("Enter a character and a string\n");
argsread = scanf ("%c %9s", &d, buf); printf
("%d arguments read\n", argsread);

printf ("Enter two floating-point numbers\n");
argsread = scanf ("%f %f", &f, &g); printf ("%d
arguments read\n", argsread);
}

```

### *Бібліотека математичних функцій*

Більшість функцій для виконання поширених математичних операцій включені до бібліотеки math.h. А такі функції як **rand** та **srand** знаходяться в stdlib.h. Слід зазначити, що у більшості програм цих функцій використовуються значення з плаваючою комою. Перелік функцій приведено у табл. 7.2

[acos](#)

[asin](#)

[atan](#)

[fabs](#)

[floor](#)

[fmod](#)

[sin](#)

[sinh](#)

[srand](#)

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 58

<a href="#"><u>atan2</u></a>	<a href="#"><u>log</u></a>	<a href="#"><u>sqrt</u></a>
<a href="#"><u>ceil</u></a>	<a href="#"><u>log10</u></a>	<a href="#"><u>tan</u></a>
<a href="#"><u>cos</u></a>	<a href="#"><u>modf</u></a>	<a href="#"><u>tanh</u></a>
<a href="#"><u>cosh</u></a>	<a href="#"><u>pow</u></a>	<a href="#"><u>chkdouble</u></a>
<a href="#"><u>exp</u></a>	<a href="#"><u>rand</u></a>	<a href="#"><u>chkfloat</u></a>

Приклад:

```
#include <math.h>
#include <stdio.h> /* for printf */

void tst_abs (void) {
    int x;
    int y;

    x = -42;
    y = abs (x);

    printf ("ABS(%d) = %d\n", x, y);
}
```

### 7.3.2 Стандартний програмний інтерфейс для Cortex-M процесорів (CMSIS)

Зі зростанням складності вбудовуваних систем дуже важливим стає забезпечення сумісності та можливості повторного використання програм. У разі можливості повторного використовувати програмне забезпечення зменшуються терміни розробки для всіх наступних проектів та скорочується час виходу продукції на ринок. Для того, щоб уникнути потреби детально вивчати технічну документацію та вручну ініціалізувати периферійні пристрої, створена універсальна стандартизована бібліотека під назвою CMSIS - Cortex Microcontroller Software Interface Standard. Як правило, виробники самі створюють власну версію цієї бібліотеки для своїх мікроконтролерів.

Бібліотека CMSIS містить у собі аступне:

- Визначення регістрів, що входять в ядро ARM, для спрощення доступу до них.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 59

- Визначення функцій для роботи з базовими периферійними пристроями всередині мікроконтролера (наприклад з таймерами).
- Також до CMSIS можуть входити функції для роботи з іншими периферійними пристроями, які виробник включив в мікроконтролер.

Використання даної бібліотеки в програмах для вбудованих систем дозволяє досить просто об'єднувати програми, що були реалізовані для інших процесорів, а також програмне забезпечення третіх сторін. Звернення до апаратних пристроїв відбувається через єдиний, стандартизований інтерфейс. Використання бібліотеки дозволяє пришвидшити процес створення програмного забезпечення, використовуючи готові рішення на рівні ініціалізації периферійних пристроїв і зосереджуючи увагу на реалізації певного застосування.

Для кращого розуміння призначення бібліотеки CMSIS на рис. 7.1 приведено узагальнену структурну схему вбудованої системи.

### 7.3.3 Бібліотека драйверів периферійних пристроїв BSP Library

Виробники мікроконтролерів на основі ARM архітектури, зазвичай, пропонують готові рішення стосовно ініціалізації периферійних пристроїв, яких є значна кількість. Готові рішення пропонуються розробникам вбудовуваних систем у вигляді бібліотек під назвою BSP Library, у тому числі і від NUVOTON.

З'ясуйте для себе структуру бібліотеки (рис. 7.2) готових рішень і додайте до проекту файли ініціалізації широтно-імпульсної модуляції (ШІМ), зокрема DrvPWM.c. Далі виконайте налаштування ШІМ на зумер і за опомогою регулятора на навчальній платі виконайте регуляцію частоти звуку, що ним відтворюється.

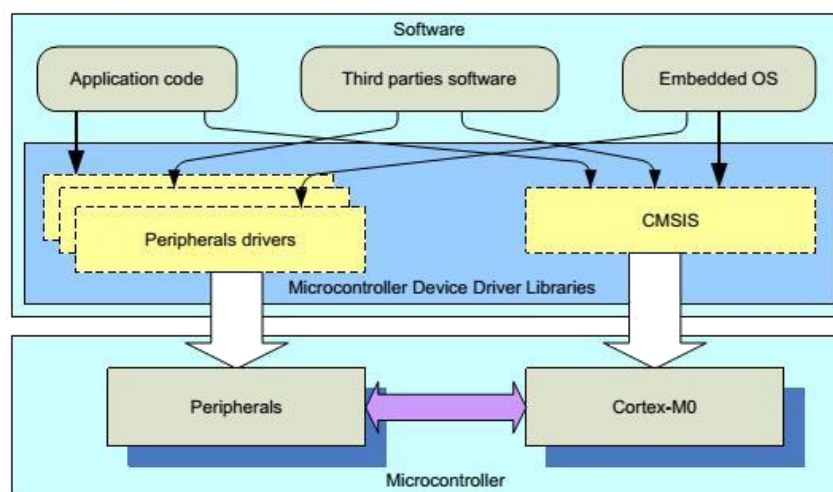


Рис. 7.1 – Роль та місце бібліотеки CMSIS у загальній схемі вбудованої системи

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 60

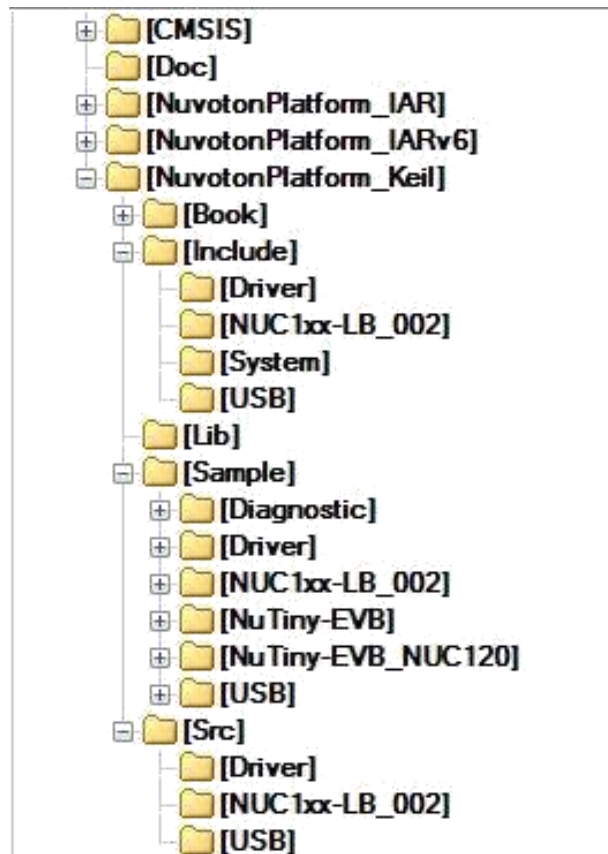


Рис. 7.2 – Структура BSP Library

### Контрольні запитання

1. Як забезпечується можливість повторного використання програм через бібліотеку CMSIS?
2. Приведіть розрахунок та вивід на екран коренів квадратного рівняння.
3. Приведіть розрахунок та вивід на екран значень за теоремою Піфагора.
4. Приведіть розрахунок та вивід на екран квадрат значення введеного числа.
5. За яким форматом буде виконуватись вивід даних у потоці через printf за шаблоном %04d.
6. Приведіть розрахунок та вивід на екран значень за теоремою синусів.
7. Приведіть розрахунок та вивід на екран логарифмічного значення введеного числа.
8. Приведіть розрахунок та вивід на екран значень за теоремою косинусів.
9. Запропонуйте програму для формування синусоїдального сигналу.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 61

10. Приведіть назви реєстрів з бібліотеки CMSIS для роботи з системним таймером SysTick.

11. За яким форматом буде виконуватись вивід даних у потоці через printf за шаблоном %4d.

12. Що саме стандартизує бібліотека CMSIS?

### Література до лабораторної роботи № 7

1. Yiu J. The Definitive Guide to the ARM Cortex-M0 / J. Yiu . – Newnes, 2011. – 518 p. – ISBN: 978-0-12-385477-3. [pp. 43-71, 224-264]

2. NUC140 Datasheet EN V3.02 [Електронний документ]. – Режим доступу: <http://www.nuvoton.com/resource-files/DA00-NUC140ENF1.pdf> – Назва з екрану. [pp. 47-151, 177-199].

3. Library Reference [Електронний документ]. – Режим доступу: [http://www.keil.com/support/man/docs/ca/ca\\_printf.htm](http://www.keil.com/support/man/docs/ca/ca_printf.htm). – Назва з екрану.

4. Вбудовані системи <http://radar.kpi.ua/downloads/>

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 62

## Лабораторна робота № 8

### ЗАСТОСУВАННЯ MATLAB SIMULINK ДЛЯ ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ STM32F4

#### 8.1 Мета та основні завдання роботи

Отримати навички реалізації уніфікованих проектів для ARM Cortex-M процесорів з метою пришвидшення розробки програмного забезпечення для вбудованих систем та можливості легкого портування проектів на ARM процесори від інших виробників та з іншою комплектацією та реалізацією периферійних пристроїв.

#### 8.1 Основні теоретичні відомості

У останні роки в університетських та інженерно-технічних колах світу спостерігається інтенсивне поширення нової комп'ютерної системи здійснення математичних розрахунків – системи Matlab.

Головні переваги «мови технічних обчислень» Matlab, які вигідно відрізняють її серед інших існуючих нині математичних систем і пакетів, полягають у такому:

1. Система Matlab спеціально створена для проведення саме інженерних розрахунків: математичний апарат, який використовується в ній, гранично наближений до сучасного математичного апарату інженера вченого і спирається на обчислення з матрицями, векторами й комплексними числами; графічне подання функціональних залежностей тут організовано у формі, яку вимагає саме інженерна документація.

2. Мова програмування системи Matlab дуже проста, близька до мови BASIC, посилює будь-якому початківцю. Вона містить усього кілька десятків операторів. Незначна кількість операторів тут компенсується великою кількістю процедур і функцій, зміст яких легко зрозумілий користувачеві з відповідною математичною й інженерною підготовкою.

3. На відміну від більшості математичних систем, Matlab є відкритою системою. Це означає, що практично всі процедури й функції Matlab доступні не тільки для використання, але й для коригування та модифікування. Matlab – система, яка може розширюватися користувачем за його бажанням створеними ним програмами й процедурами (підпрограмами). Її легко пристосувати до вирішення задач певних класів.

4. Дуже зручною є можливість використовувати практично всі обчислювальні можливості системи в режимі надзвичайно потужного науково-го калькулятора. У той же час можна складати власні окремі програми з метою багаторазового їх використання

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 63

для досліджень. Це робить Matlab незамінним засобом проведення наукових розрахункових досліджень.

5. Останні версії Matlab дозволяють легко інтегрувати її з текстовим редактором Word, що робить можливим використання при створенні текст-тових документів обчислювальних і графічних можливостей Matlab.

Нові версії Matlab дозволяють за допомогою спеціальних бібліотек програмувати різні мікроконтролери, такі як Texas Instrument, STM32, Arduino. Ці бібліотеки необхідно коректно встановити в середовищі Matlab Simulink. Далі їх можна знайти в загальній бібліотеці Matlab Simulink. Використання спеціальних бібліотек для програмування мікроконтролера дозволяє значно полегшити процес програмування. Тобто програмування стає наочним і візуальним, що дуже зручно для інженера.

Після компіляції зібраної моделі генерується програмний код на мові програмування C. За необхідності програмний код можна модифікувати. Також створюється бінарний файл створеної програми, який згодом завантажуватиметься в мікроконтролер. Таким чином можна створювати програми для мікроконтролера за допомогою створення візуальної математичної моделі. Знання складних мов програмування, таких як C, при програмуванні за допомогою середовища моделювання Matlab не потрібно. Тому програмування мікроконтролерів за допомогою Matlab Simulink є актуальним.

#### ***Алгоритм налаштування Matlab Simulink для програмування мікроконтролера STM32F4Discovery***

Загальний алгоритм налаштування середовища моделювання Matlab Simulink наведено на рис. 8.1.

Для встановлення необхідних бібліотек-доповнень для здійснення програмування мікроконтролера STM32F4Discovery необхідно на панелі інструментів у меню Add-Ons вибрати пункт Get Hardware Support Packages (рис. 8.2).

Для програмування мікроконтролера STM32F4Discovery зі списку пропонованих пакетів підтримання необхідно вибрати пункт STMicroelectronics STM32F4Discovery (рис. 8.3).

У вікні можна вибрати встановлення пакетів підтримання з Інтернету, папки розташування файлів бібліотек, завантаження або видалення пакетів підтримання. У цьому випадку вибрано Install from Internet (встановити з Інтернету).

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 64



Рис. 8.1 – Алгоритм налаштування Matlab Simulink для програмування мікроконтролера STM32F4discovery

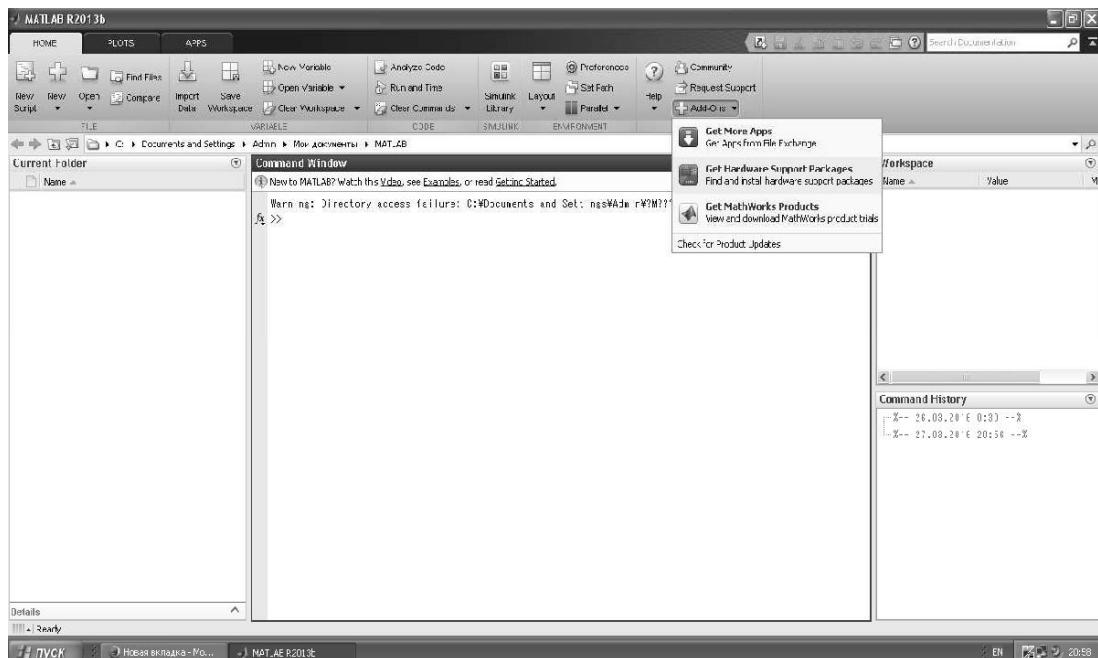


Рис. 8.2 – Початок встановлення необхідних бібліотек



Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 65

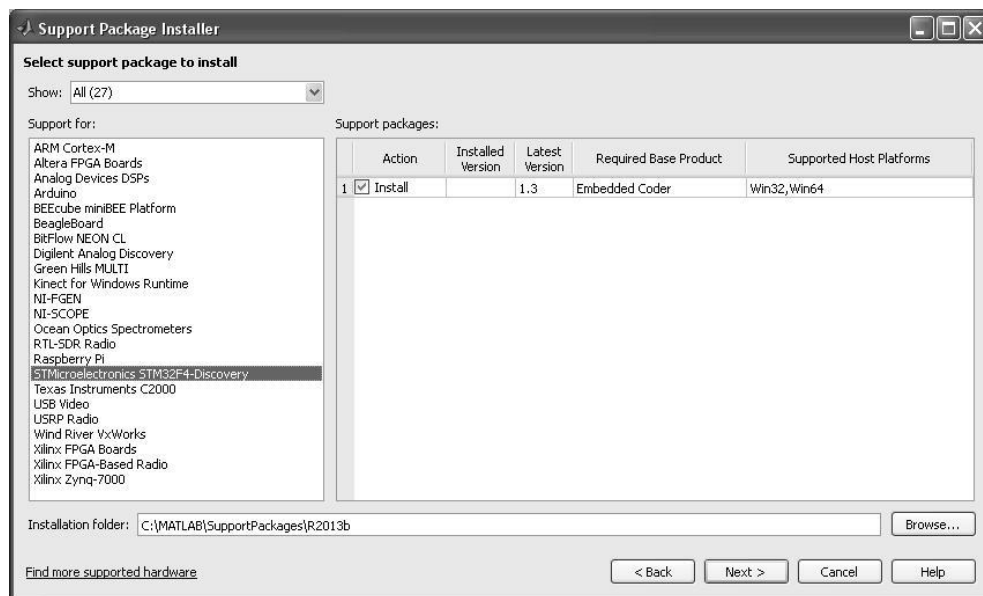


Рис. 8.3 – Вибір пакетів підтримання для мікроконтролера STM32F4Discovery

Для того щоб продовжити установлення пакетів підтримання, необхідно натиснути Log In та у вікні ввести власний email і пароль від облікового запису, зареєстрованого на сайті [www.mathworks.com](http://www.mathworks.com).

У разі якщо відсутній обліковий запис, його можна створити, натиснувши Create an account в наведеному на цьому слайді вікні або на сайті [www.mathworks.com](http://www.mathworks.com). Після створення облікового запису необхідно ввести власний email і пароль і натиснути Log In. Після чого користувач отримує можливість ознайомитися з умовами ліцензійної угоди допоміжного про-грамного забезпечення від MathWorks.

Далі користувачеві буде подано список встановлюваного програмно-го забезпечення сторонніх виробників і звідки воно буде завантажено та посилання на ліцензійні угоди програмного забезпечення. При продовженні установлення відбувається прийняття ліцензійних угод автоматично.

У наступному вікні буде вказано, які пакети підтримання потрібно встановити і шлях їх установлення. Необхідно підтвердити установлення, натиснувши клавішу Install.

Після цього відкриється вікно, у якому буде відображатися процес завантаження програмного забезпечення. Необхідно дочекатися завершення завантаження й установлення.

Далі з'явиться вікно, яке підтверджує успішне завершення першого етапу установлення пакетів підтримання й пропонує продовжити установлення.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 66

У наступному вікні буде запропоновано вибрати встановлюваний пакет підтримання STMicroelectronics STM32F4Discovery.

Після буде наведено список необхідного додаткового програмного забезпечення. При натисканні клавіші download відкриється сайт, з якого можна завантажити необхідне програмне забезпечення. Для можливості його завантаження необхідна попередня реєстрація на цьому сайті.

Після завантаження необхідного програмного забезпечення в наступному вікні необхідно буде вказати шлях до скачаних файлів і підтвердити установлення, натиснувши клавішу Validate.

Потім буде запропоновано перевірити правильність встановлення програмного забезпечення, його версії та шляхи до скачаних файлів. Розташування блоку Target Setup наведено на рис. 8.4.

Налаштування програматора Target Setup для створення програми для мікроконтролера STM32F4Discovery наведено на рис. 8.5,8.6.

Розташування блоку цифро-аналогового перетворювача в бібліотеці компонентів Matlab Simulink Simulink Library Browser наведено на рис. 8.7.

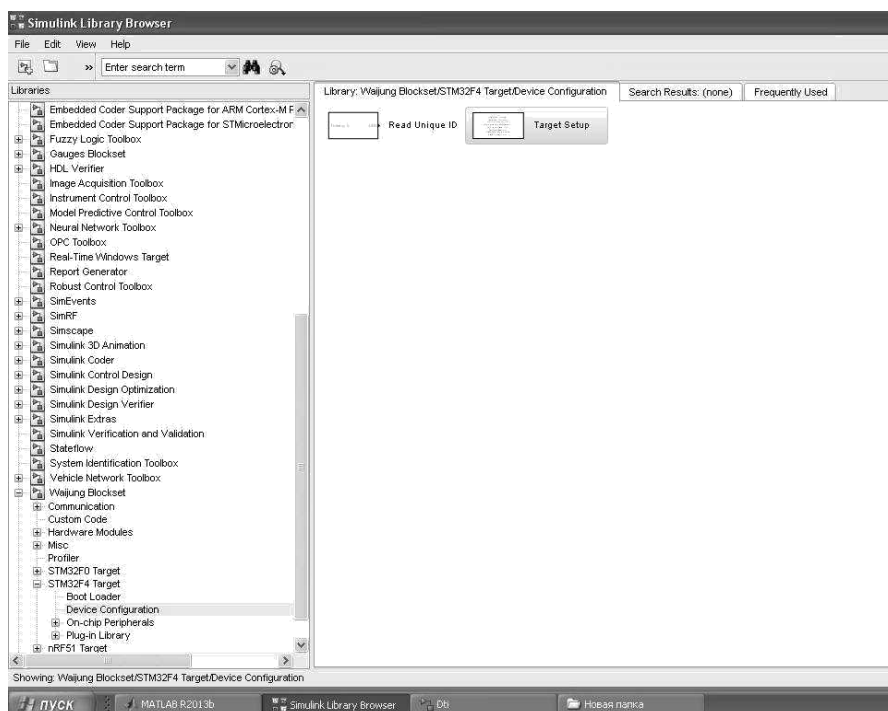


Рис. 8.4 – Розташування блоку Target Setup

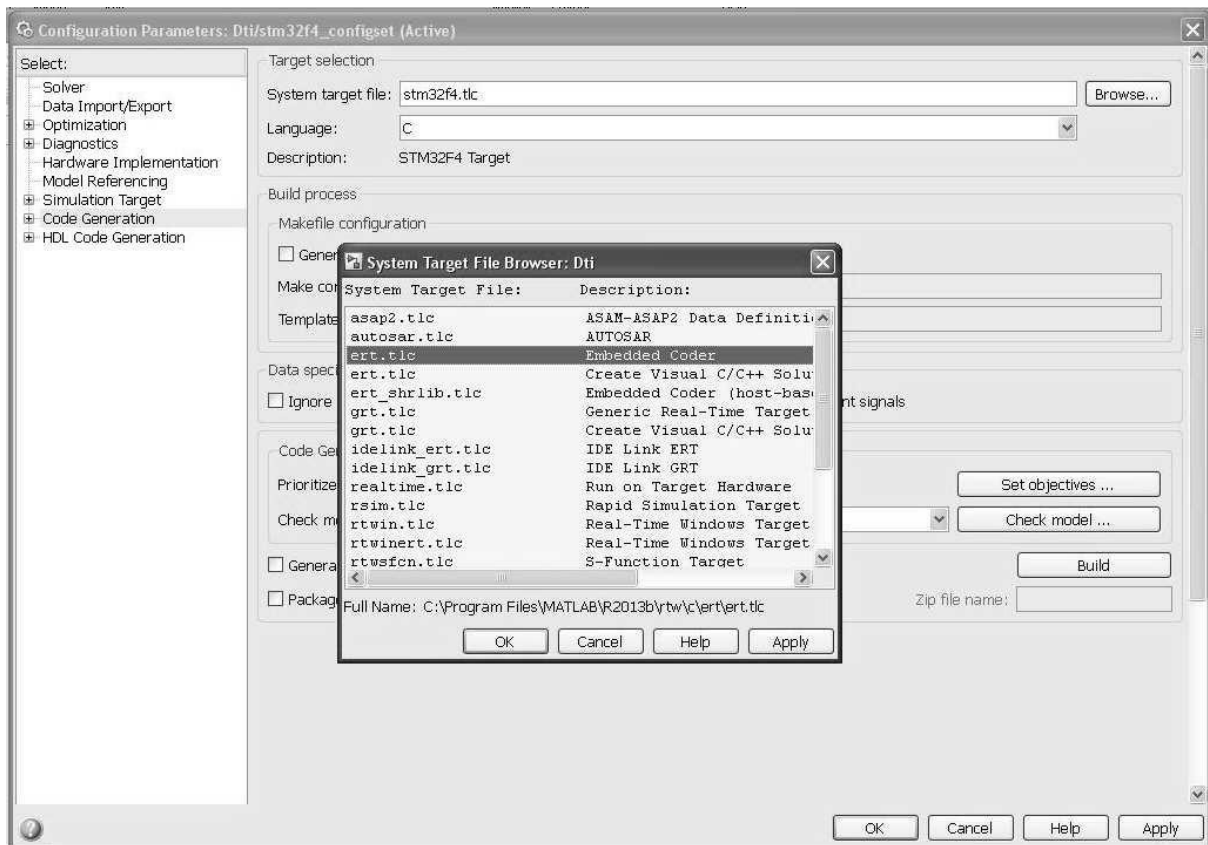


Рис. 8.5 – Налаштування блоку Target Setup

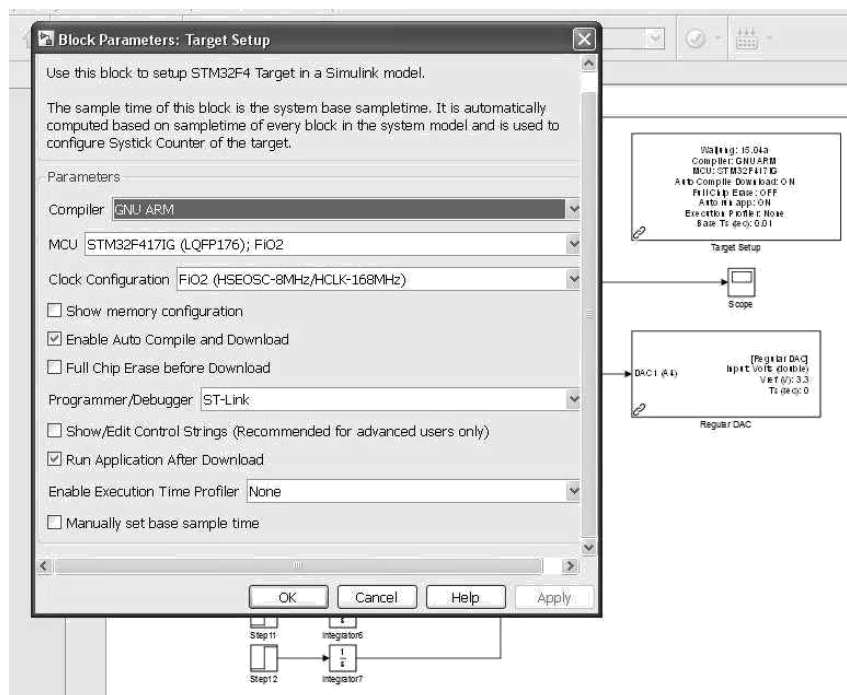


Рис. 8.6 – Налаштування блоку Target Setup у моделі Matlab Simulink

<b>Житомирська політехніка</b>	<b>МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015</b>	<b>Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020</b>
	<i>Екземпляр № 1</i>	<i>Арк 64 / 68</i>

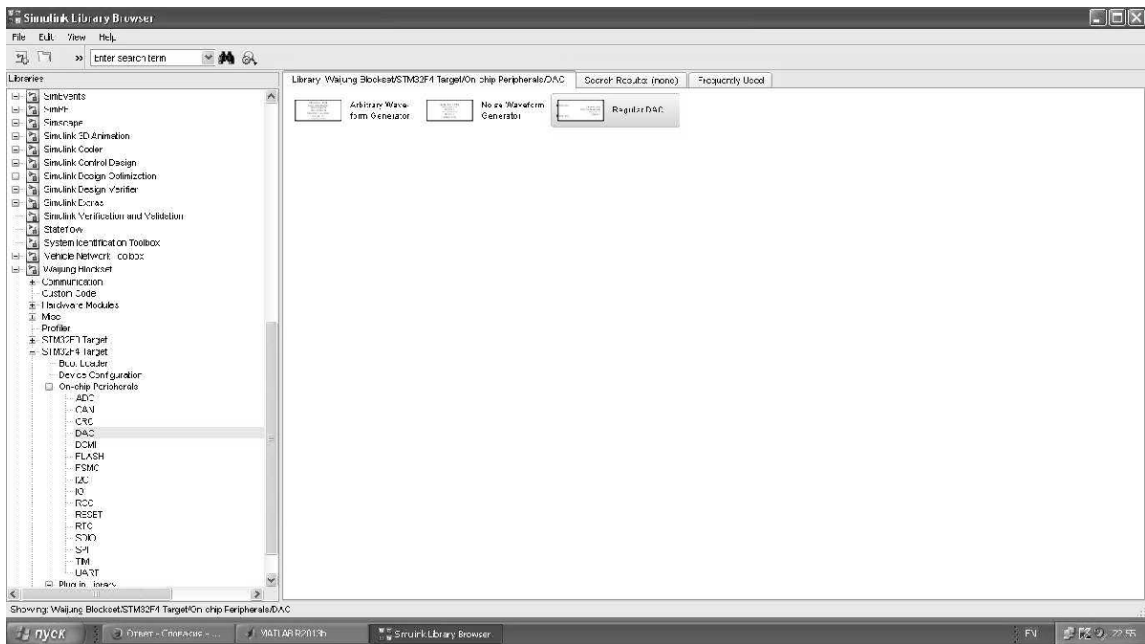


Рис. 8.7 – Розташування блоку Regular DAC у бібліотеці Simulink Library Browser

Налаштування блоку цифро-аналогового перетворювача наведено на рис. 8.8.

Результати моделювання роботи цифро-аналогового перетворювача у Matlab Simulink і осцилограма вихідного сигналу цифро-аналогового перетворювача на виході порту A PA4 мікроконтролера STM32F4Discovery наведено на рис. 8.9 (а, б), відповідно.

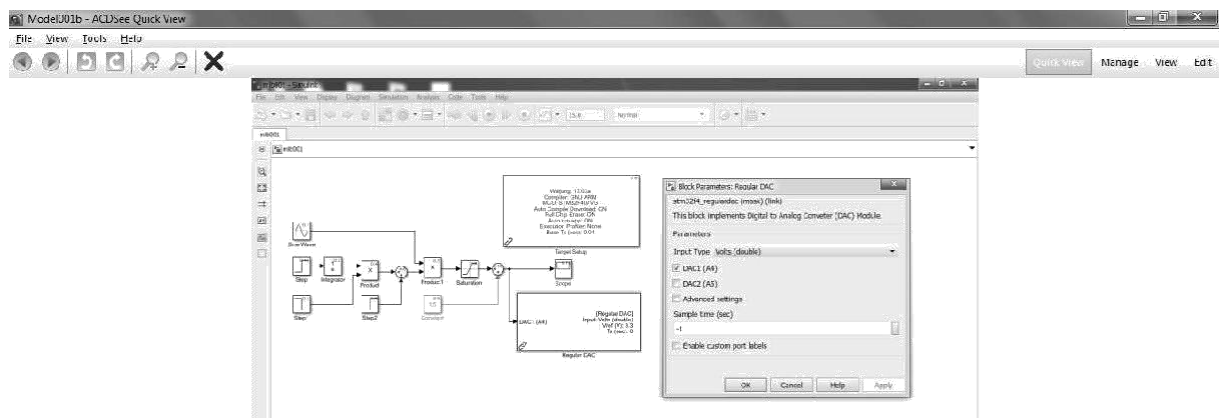
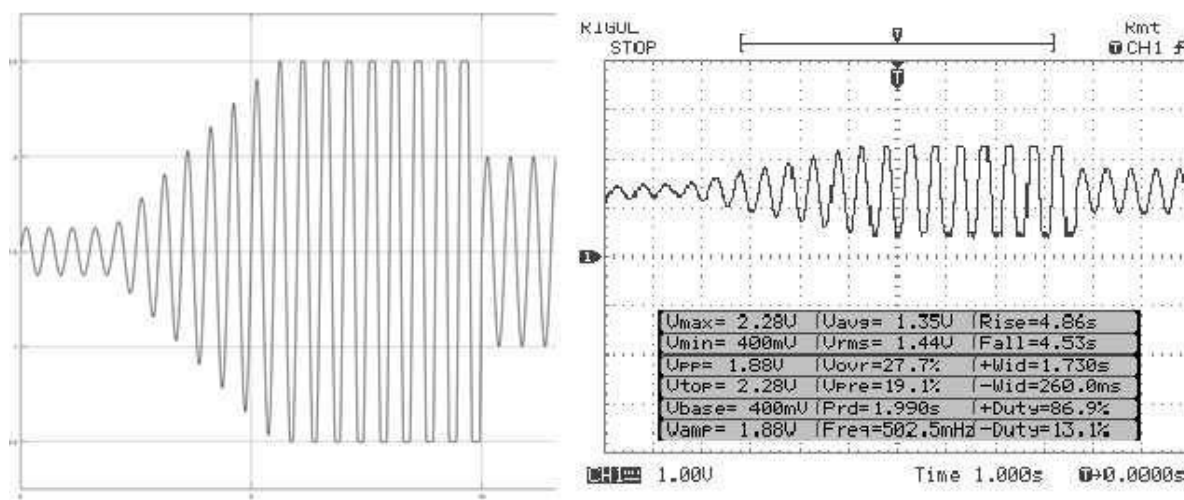


Рис. 8.8 – Налаштування блоку Regular DAC у моделі Matlab Simulink

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідас ДСТУ ISO 9001:2015	Ф-20.09- 05.02/2/152.00.1Б/ОК14- 2020
	Екземпляр № 1	Арк 64 / 69



а

б

Рис. 8.9 – Графік сформованого сигналу на виході цифро-аналогового перетворювача в Matlab Simulink (а), осцилограма сформованого сигналу на виході РА4 мікроконтролера STM32F4Discovery (б)

Таким чином, використовуючи Matlab Simulink для програмування мікроконтролерів, можна розробляти програми будь-якої складності без знання мови програмування С. Створена модель перетвориться системою Matlab Simulink на С-код у вигляді окремих модулів. Також створюється бінарний файл прошивки з розширенням BIN. Цей файл можна завантажити в мікроконтролер за допомогою спеціалізованої утиліти для прошивання, наприклад ST-Link.

Однак для оптимізації скомпільованого коду, повного використання можливостей мікроконтролера STM32F4Discovery знання мови програмування С не тільки бажано, але й необхідно.

### Література до лабораторної роботи № 8

1. Квашнін, В.О. Програмування та застосування мікроконтролерів STM32F4Discovery: монографія / В. О. Квашнін, А.В. Бабаш, В.В. Квашнін. – Краматорськ : ЦТПІ «Друкарський дім», 2017. – 143 с.